1989年8月1日発行(毎月1回1日発行)第8巻8号通巻88号　昭和58年11月2日第三種郵便物認可

ISSN 0910-7614

パーソナルコンピュータ・マガジン
MZシリーズ,X1/turbo,X68000&ポケコン

Oh!X

オー!エックス 定価560円

特集1
X1プログラミングガイドブック

PCGの基礎から奥義まで/超高速ラインルーチン
パズルゲームThe Rescuer

特集2
3Dグラフィックの深淵へ

スキャンラインZバッファによる映像表現
3Dモデリングを考える
サイクロンExpress試用レポート

新連載
(で)のショートプロぱーてぃ

MZ-2500
グラフィックエディタ作成講座/MIDI入門(3)

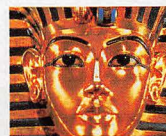
X68000
X-BASIC調理実習/マシン語プログラミング
C調言語講座PRO-68K/DōGA・CGA講座

S-OS
CP/M用ファイルコンバータ

THE SOFTOUCH
ニュージーランドストーリー/第4のユニット3
ソフトでハードな物語2/Might and MagicⅡ

猫とコンピュータ/知能機械概論
マシン語カクテル in Z80's Bar

8
AUG. 1989

夢のつづきを語ろう。

いま、ヒューマンインターフェイスなX68マインド ——

「スペックはすべてを語り尽くせない」、このパラドックスは、パーソナルな機器としてのコンピュータの難しさを端的に表現しています。〝クリエイティブマインド〟や〝ヒューマンインターフェイス〟は、ハードウェアレベルにしろ、アプリケーションレベルにしろ、スペック上で全てがわかるというものでもありません。そうした意味からも、単にハードウェアとしての32ビットには疑問をはさむ余地がありすぎるのも、また事実です。何を実現したいのか。X68000は、少なくともその可能性を提示し得た数少ないマシンのなかのひとつといえます。パーソナルデータとは何か、ビジュアルインターフェイスの意味、開発当初のコンセプトは、風化するどころか、いま現実となってますますクローズアップされてきています。かゆいところに手が届く、そんなヒューマンインターフェイスのひとつひとつをクリアーしていく、血の通ったテクノロジーに新世代マシンのイメージがふくらんできませんか。

●そのヒューマンインターフェイスを推し進めて、X68000のシステムパフォーマンスをさらに高めたものとしてHuman68k ver.2.0があります。EXPERT, PRO両シリーズに搭載されたこのOSは、従来通りのマルチウインドゥやアイコンを駆使したビジュアルシェルによるフレンドリーな操作環境に、将来性をみこした数々の処理機能を装備。インテリジェントな環境を実現しています。まず、マルチタスクに近い処理環境を提供するバックグラウンド処理の実現。バックグラウンドで動作するサンプルとして標準でTIMERコマンドを用意し、ある処理を行いながら指定時刻にAD PCMファイルを再生させたり、1画面分のメモ程度のファイルを画面に表示させることができます。次に、これからのワークステーション環境に必要なネットワーク処理にそなえ、ファイルの共有化とロックや仮想ドライブ対応などをサポートしています。さらに、キー入力や編集を効率的に行えるヒストリデバイスドライバの採用。拡張されたヒストリ機能に加え、コマンドを別の名前で定義するエイリアス機能、キー入力の履歴からユーザーが自由に複数行を登録しておき、連続実行できる簡易バッチ機能などを装備しています。その他、約2倍にスピードアップされたファイルアクセス(V1.0比)、大容量ファイルアクセスを可能にし、光磁気ディスクなど将来の大容量メディアへの対応、メニュー方式の簡単なキー操作で外部コマンドやアプリケーションを実行できるMENUコマンドの装備など、さらに高い次元へと進化した処理機能とヒューマンインターフェイス。まさにワークステーションと呼ぶにふさわしいシステムパワーを実現しました。●また、日本語処理に対してもヒューマンインターフェイスを追求した日本語フロントエンドプロセッサver.2.0を搭載。約2倍にスピードアップ(V1.0比)された変換速度をはじめ、キー割り付けの自由設定、カーソル位置での文字入力や変換など、フレンドリーなオペレーティングを実現する操作環境をサポートしています。

〈共通特長〉●プロセッサの未来を先取りした68000搭載●テキスト、グラフィック、スプライトの3画面を独立させた独自のメモリアーキテクチャ●1024×1024ドット(最大表示エリア768×512ドット)、高品位な金属の質感までも自然に表現しうる65,536色同時発色(512×512ドット時)の高解像度自然色グラフィックス●16×16ドットの緻密なキャラクタを駆使できるスプライト機能(水平32スプライト、1画面128スプライト、65,536色中16色)●ステレオFM音源、ADPCM搭載●オートロード、オートイジェクトメカ採用。インテリジェントな1Mバイト5"FDD2基搭載。●蓄積されたソフトが利用できるX68000シリーズとソフトコンパチ。

EXPERTシリーズ

●高密度実装を象徴するフォルム、マンハッタンシェイプ●新たな領域をひらく3Mバイトの大容量メモリを標準装備、メインメモリは標準で2Mバイト、最大12Mバイトまで拡張可能●プロフェッショナルなクリエイティブワークに対応する40Mバイトハードディスク搭載(CZ-612C)*●マウス・トラックボール標準装備●日本語入力にスムースに対応するASCII準拠フルキーボード

*CZ-602Cには、本体内に内蔵できる増設用の40Mバイトハードディスクドライブ(標準価格120,000円税別・取付費別)をサポート。

PROシリーズ

●意表をつくボディコンストラクション、高度な実装技術に裏付けられた洗練と信頼性の新しいスタンダードフォルム●高度なシステム化への対応を考慮した拡張I/Oスロット4スロット標準装備●プロニーズの大容量ファイルに対応した40Mバイトハードディスク搭載(CZ-662C)*●2Mバイトの大容量メモリを標準装備●マウス標準装備●ワイドスケールのフルキーボード採用

*CZ-652Cには、本体内に内蔵できる増設用の40Mバイトハードディスクドライブ(標準価格120,000円税別・取付費別)をサポート。

選べる3タイプのディスプレイをサポート

15型カラーディスプレイテレビ(ドットピッチ0.39mm) CZ-602D-GY(グレー)・-BK(ブラック) 標準価格 99,800円(チルトスタンド同梱・税別)
15型カラーディスプレイテレビ(ドットピッチ0.31mm) CZ-612D-GY(グレー)・-BK(ブラック) 標準価格119,800円(チルトスタンド同梱・税別)
14型カラーディスプレイ (ドットピッチ0.31mm) CZ-603D-GY(グレー)・-BK(ブラック) 標準価格 84,800円(チルトスタンド同梱・税別)

●写真左はCZ-612C-BK+CZ-612D-BK、写真右はCZ-652C-GY+CZ-603D-GY

EXEリーダーズ「カップ」プレゼント実施中

●いま、EXE会員よりご紹介のお客様がEXEショップでX68000シリーズを購入されますと、EXE会員にEXEリーダーズ「カップ」をプレゼントします。詳しくはEXEショップにお問い合わせください。
●また、X68000シリーズをご購入のお客様は、ぜひEXEクラブにご入会ください。

本広告に掲載しております商品および役務の価格には消費税は含まれておりませんので、ご購入の際、消費税をお支払い下さい。

シャープ株式会社 ●お問い合わせは…シャープ㈱電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)
電子機器事業本部テレビ事業部第4商品企画部 〒162 東京都新宿区市谷八幡町8番地 ☎(03)260-1161(大代表)

Oh!X

CONT

〈スタッフ〉
●編集長/前田 徹 ●副編集長/永野 仁 ●編集/植木章夫 石塚康世 高野庸一 ●協力/有田隆也 中森 章 清水和人 後藤貴行 林 一樹 荻窪 圭 岡本浩一郎 毛内俊行 吉田賢司 影山裕昭 相馬英智 古村 聡 村田敏幸 丹 明彦 三沢和彦 長沢淳博 宮島 靖 金子俊一 ●カメラ/杉山和美 ●イラスト/永沢しげる 山田晴久 小栗由香 ●アートディレクター/島村勝頼 ●レイアウト/元木昌子 AD GREEN ●校正/手塚喜美子 千野延明

表紙絵: Moto Noriyuki

■広告目次

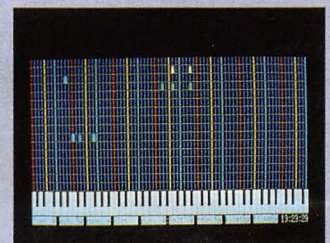
1989 AUG.
8

E N T S

特集1 パズルゲーム The Rescuer

特集2 Zバッファアルゴリズム(後編)

ニュージーランドストーリー

第4のユニット3・デュアルターゲット

MZ-2500MIDI用鍵盤表示システム

(で)のショートプロぱーてぃ

SHARP

クリエイティブマインドあふれる周辺機器が

CZ-600C/601C/611C/602C/612C

ディスプレイ関連

カラーディスプレイテレビ

15型カラーディスプレイテレビ
CZ-602D-GY・-BK
標準価格 99,800円（税別）
（チルトスタンド同梱）
NEW

15型カラーディスプレイテレビ
CZ-612D-GY・-BK
標準価格 119,800円（税別）
（チルトスタンド同梱）
NEW

カラーディスプレイ

21型カラーディスプレイ
CU-21CD
標準価格 139,800円（税別）

14型カラーディスプレイ
CZ-603D-GY・-BK
標準価格 84,800円（税別）
（チルトスタンド同梱）

チューナー

RGBシステムチューナー
CZ-6TU-GY・-BK
標準価格 33,100円（税別）
（リモコン付）

CRTフィルター

高性能CRTフィルター
BF-68PRO
標準価格 19,800円（税別）
（CZ-600D/602D/612D/603D用）

アートツール

画像入力

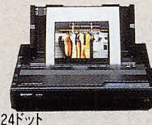
カラーイメージスキャナ※1
CZ-8NS1
標準価格 188,000円（税別）

スキャナ用パラレルボード
CZ-6BN1
標準価格 29,800円（税別）

映像入力

カラーイメージユニット
CZ-6VT1
CZ-6VT1-BK
標準価格 69,800円（税別）

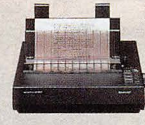
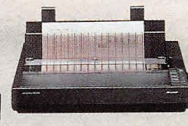
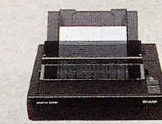
プリンタ

カラープリンタ

24ドット
熱転写カラー漢字プリンタ
CZ-8PC3
標準価格 65,800円（税別）
（信号ケーブル同梱）

48ドット
熱転写カラー漢字プリンタ
CZ-8PC4
標準価格 99,800円（税別）
（信号ケーブル同梱）
NEW

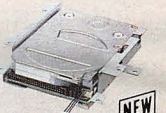
カラービデオプリンタ

カラービデオプリンタ
★CZ-6PV1
標準価格 198,000円（税別）
（信号ケーブル同梱）

ドットプリンタ

24ピン漢字プリンタ（80桁）
CZ-8PK7
標準価格 122,000円（税別）
（信号ケーブル同梱）

24ピン漢字プリンタ（136桁）
CZ-8PK8
標準価格 152,000円（税別）
（信号ケーブル同梱）

24ピン漢字プリンタ（80桁）
CZ-8PK9
標準価格 89,800円（税別）
（信号ケーブル同梱）

ファイル

ハードディスク

ハードディスクユニット（20MB）
CZ-620H
標準価格 178,000円（税別）

増設用ハードディスクドライブ（40MB）
CZ-64H
標準価格 120,000円（税別）
（取付費別）
NEW
※取付に関してはシャープお客様ご相談窓口にてご相談ください。

※1 ご使用に際しては、カラーイメージスキャナCZ-8NS1に同梱のRS-232Cケーブルで接続するか、より高速のパラレルデータ伝送を行う場合、別売のスキャナ用パラレルボードCZ-6BN1標準価格29,800円（税別）で接続してください。
※2 CZ-652C、662Cをお持ちの方は包装箱の表示形名CZ-6BE1Aの右横にⒶマーク表示のあるものをお買い求めください。

XVI・XVI turbo シリーズ用 周辺機器

標準価格は税別です。

カラーディスプレイ		
●21型カラーディスプレイ※1	CU-21CD	139,800円

映像・画像入力編集装置		
●カラーイメージスキャナ	CZ-8NS1	188,000円
●カラーイメージボードII	CZ-8BV2	39,800円
●立体映像セット	★CZ-8BR1	29,800円
●パーソナルテロッパ※2	CZ-8DT2	44,800円

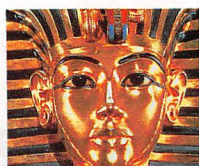
FM音源		
●ステレオタイプFM音源ボード	CZ-8BS1	23,800円

スピーカー（2本1組）標準装備、ミュージックツール同梱

プリンタ		
●24ピン漢字プリンタ（80桁）	CZ-8PK7	122,000円
●24ピン漢字プリンタ（136桁）	CZ-8PK8	152,000円
●24ピン漢字プリンタ（80桁）	CZ-8PK9	89,800円
●24ドット熱転写カラー漢字プリンタ	CZ-8PC3	65,800円
●48ドット熱転写カラー漢字プリンタ	CZ-8PC4	99,800円
●カラービデオプリンタ	★CZ-6PV1	198,000円

ファイル		
●ミニフロッピーディスクユニット（2HD・2D）※3	★CZ-520F	118,000円
●ミニフロッピーディスクユニット（2D）	★CZ-502F	99,800円

シャープ株式会社
●お問い合わせは…シャープ㈱電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221（大代表）

X68000をサポート。

シャープペリフェラルファミリー X68000

CZ-652C/662C

ボード

拡張メモリ

1MB増設RAMボード
(CZ-600C用)
CZ-6BE1
標準価格 35,000円(税別)

1MB増設RAMボード※2
(CZ-601C/611C/652C/662C用)
CZ-6BE1A
標準価格 38,000円(税別)

2MB増設RAMボード※3
CZ-6BE2
標準価格 79,800円(税別)

4MB増設RAMボード※3
CZ-6BE4
標準価格 138,000円(税別)

インターフェイス

ユニバーサルI/Oボード
CZ-6BU1
標準価格 39,800円(税別)

GP-IBボード
CZ-6BG1
標準価格 59,800円(税別)

増設用RS-232Cボード
(2チャンネル)
CZ-6BF1
標準価格 49,800円(税別)

数値演算プロセッサ

数値演算プロセッサボード
CZ-6BP1
標準価格 79,800円(税別)

FAX

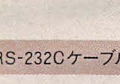
FAXボード
CZ-6BC1
標準価格 79,800円(税別)

MIDI

MIDIボード
CZ-6BM1
標準価格 26,800円(税別)

ネットワーク

モデム

モデムユニット※4
CZ-8TM2
標準価格 49,800円(税別)
(RS-232Cケーブル同梱)

RS-232Cケーブル

RS-232Cケーブル
(平行接続型)
CZ-8LM1
標準価格 7,200円(税別)

RS-232Cケーブル
(クロス接続型)
CZ-8LM2
標準価格 7,200円(税別)

入力

NEW

インテリジェントコントローラ
CZ-8NJ2
標準価格 23,800円(税別)

トラックボール
CZ-8NT1
標準価格 13,800円(税別)

マウス
CZ-8NM2A
標準価格 6,800円(税別)

ジョイカード
CZ-8NJ1
標準価格 1,700円(税別)

その他

拡張スロット

拡張I/Oボックス(4スロット)
(CZ-600C/601C/611C/602C/612C用)
CZ-6EB1
CZ-6EB1-BK
標準価格 88,000円(税別)

スピーカー

アンプ内蔵
スピーカーシステム(2本1組)
AN-S100
標準価格 36,600円(税別)

システムラック

システムラック
CZ-6SD1
標準価格 44,800円(税別)

※3 ご使用に際しては、あらかじめ別売の1MB増設RAMボードCZ-6BE1 標準価格35,000円(税別・CZ-600C用)、CZ-6BE1A 標準価格38,000円(税別・CZ-601C、CZ-611C、652C、662C用)を増設してください。
※4 モデムユニットCZ-8TM2に同梱のソフトはX1/X1ターボシリーズ用です。

●ミニフロッピーディスクユニット(2D・1ドライブ)	CZ-503F	49,800円
●増設用ミニフロッピーディスクドライブ(2D)※4	CZ-53F-BK	19,800円

拡張ボード・その他		
●モデムユニット(300/1200ボー)	CZ-8TM2	49,800円
●320KB外部メモリ	CZ-8BE2	29,800円
●RS-232C・マウスボード※5	CZ-8BM2	19,800円
●フロッピーディスクインターフェイス※6	CZ-8BF1	14,800円
●JIS第1水準漢字ROM※7	CZ-8BK2	19,800円

●RS-232C用ケーブル(平行接続型)	CZ-8LM1	7,200円
●RS-232C用ケーブル(クロス接続型)	CZ-8LM2	7,200円
●拡張I/Oボックス	CZ-8EB3	33,800円
●RFコンバータ※8	AN-58C	2,980円
●インテリジェントコントローラ	CZ-8NJ2	23,800円
●マウス	CZ-8NM2A	6,800円
●トラックボール	CZ-8NT1	13,800円
●ジョイカード	CZ-8NJ1	1,700円
●チルトスタンド※9	CZ-6ST1-E・-B	5,800円

●高性能CRTフィルター※10	BF-68PRO	19,800円
●スキャナ用パラレルボード※11	CZ-8BN1	27,800円

●品番中の−表示は、B〈ブラック〉・E〈オフィスグレー〉を示します。※1 X1ターボZシリーズ用 ※2 CZ-862Cには接続できません ※3 X1ターボシリーズ用 ※4 CZ-830C用 ※5 X1シリーズ用 ※6 CZ-850CでCZ-520Fを使用する場合に必要 ※7 CZ-800C、801C、802C、803C、811C、820C用 ※8 CZ-820C、822C、830C用 ※9 CZ-600D、880D、830D用 ※10 CZ-600D、602D、612D、880D、830D用 ※11 CZ-8NS1用 ●接続等の説明につきましては、周辺機器総合カタログをご参照ください。

★印の商品は在庫僅少です。

本広告に掲載しております商品および役務の価格には消費税は含まれておりませんので、ご購入の際、消費税をお支払い下さい。

電子機器事業本部テレビ事業部第4商品企画部 〒162 東京都新宿区市谷八幡町8番地 ☎(03)260-1161(大代表)

SHARP

"アート"と呼べる高水準のソフトウェアが

必要なとき、いつでも使える、サッと呼び出せる。メモリ常駐型の便利ツール。いまトレンディなステーショナリーソフトウェア。

文房具感覚で使えるサポートツール

Stationery PRO-68K

CZ-240BS　　8月発売予定

パソコンを本当に道具として、現実的に使いこなしたい、そんなユーザーのためのソフトウェアが登場しました。「Stationery PRO-68K」、それはパソコンの使い勝手を飛躍的に向上させる便利ツールです。

●他のソフトウェアを実行中でも呼び出して使えるメモリ常駐型のソフトウェア。

使い方は簡単、他のアプリケーションを起動する前に、この「Stationery PRO-68K」を一度起動するだけ。これで他のアプリケーション実行中にも、「メモ」や、「スケジュール」、「住所録」など、「Stationery PRO-68K」の持つ機能がワンタッチで使えます。いちいちアプリケーションを終了させることもありません。

●シャープ電子手帳のデータをX68000で入力、編集。

パソコン上で入力したデータを電子手帳の「電話帳」、「スケジュール」、「メモ」へデータを送信したり、逆に電子手帳側からデータを受信して編集することができます。(別売の通信ケーブル)が必要です。

メモ ●先頭/最終ページへのジャンプ ●文字列の置換、検索 ●ファイルの読み込み/書き出し、などの機能をもつ小型エディタ。

カレンダー ●任意の半年分のカレンダー表示可能。

スケジュール ●日付/時間/メモという3つの項目を持つ表形式 ●入力したデータは日付順に自動整列 ●先頭/最終ページへのジャンプや文字列検索機能を装備。

住所録 ●氏名/索引/電話/住所の4つの項目をもつ表形式 ●先頭/最終ページへのジャンプ、項目ごとの整列、文字列検索、重複データ消去、ファイルの読み込み/書き出し、などの機能を装備。

シャープ株式会社 ●お問い合わせは…シャープ㈱電子機器事業本部テレビ事業部 第4商品企画部 〒162 東京都新宿区市谷八幡町8番地 ☎(03)260-1161(大代表)へ。

資料請求券 CZソフト oh!X 8係

株式会社 キャスト
〒158 東京都世田谷区等々力2-1-13
TEL.03-705-0656 FAX.03-705-5224

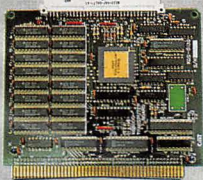
CasT

C-TRACE TOWNS	¥68,000
★C-TRACE 98FB	¥98,000
C-TRACE 68(X68000対応)	¥68,000
C-TRACE 98DRY(PC-9801対応)	¥68,000
C-TRACE 98+(PC-9801対応)	¥198,000
C-TRACE NEWS(SONY)	¥380,000
★C-TRACE 98TP	¥610,000
★C-TRACE 68TP	¥610,000

表示価格に消費税は含みません。
★の製品は店頭販売いたしません。直接当社までお申し込みください。

▲このCGは、千葉県山川秀幸さんの作品です。

この夏、うれしいはやさです。

新登場、超高速レイトレーシング
C-TRACE TP新発売!!
トランスピューターの使用により、動作速度はグーンとアップ。
PC-9801VMの80倍、X68000のなんと、170倍で新登場です。
この速さと美しさをぜひお試しください。

★「一光・美奈子のC-TRACE教室」開講。

7月より毎週金曜日、市ヶ谷Sharp(株)内OAショールームにて開催中。詳しくは、(株)キャストまで。
講師:長谷川一光、宮嶋美奈子(CG作家)

★C-TRACEによるTVCM作品集(VHSのみ)をお分けいたします。
代金1,000円を現金書留にてお送りください。送料着払いにて発送いたします。

プロのための3次元コンピューター グラフィックス

C-TRACE

商標登録申請中

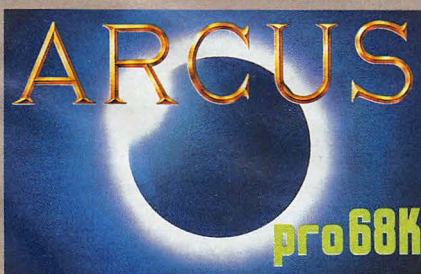
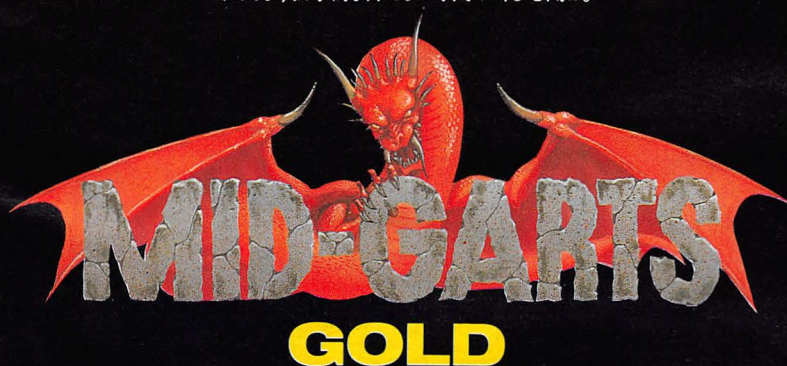
皇帝シヴァを倒すため、
剣をにぎりしめた！
■適応機種：X68000
定価 ¥9,800
(5"2HD 4枚組)
好評発売中！
トリトーン ファイナル
TRITORN FINAL
●爆発的人気を誇る「トリトーン」の決定版！
X68000専用バージョン、初のアクション、R.P.G.。
●「トリトーン2」オリジナルシナリオを超える数々
の謎とイベント。そして、増大なアイテム数と魔法数。
●魔法効果をよりビジュアルに表現させた抜群のアクション性能。
●オールタイム・フルカラー(512×512、256色)二重マップクロールのX68000専用
グラフィック。
●ADPCM(音声合成)搭載の豊富な効果音。
●ミュージックファン待望のオリジナルB.G.M.。
PRESENTED BY XAIN
僕は夏の夜に雪国をさまよい歩く…。
■適応機種：X68000
定価 ¥8,800
(5"2HD 3枚組)
8月下旬発売予定
雪の国クルージュ
●「クルージュ」オリジナルシナリオに多数のイベントを追
加し(X68000版専用バージョン)アクション・アドベンチャー・
ゲーム新登場！
●目を見張る素晴らしいグラフィックに広大なマップ(512×512、256色)
●更に奥深い謎の連続、そして次々と明らかになる新事実！
●ストーリー重視の新しいゲームジャンルのアクション・アドベンチャー。
●多数のオリジナルB.G.Mに加えて、ADPCM(音声合成)による豊富な効果音。
グラフィック・
スプライト・
キャラクターエディター
G-ツール
■適応機種：X68000
■アセンブラ・C言語でオリジナルゲーム
(X68000の機能をフルに活かした)をこれか
ら制作しようと思う人に使っていただきたい本
格派ツールの登場です。
●本格派スプライトエディター
●フルカラー・グラフィックエディター
●キャラクターエディター
(マウス対応・全グラフィックモード対応)
予価 ¥28,000
発売日未定
※表示価格には消費税は含まれておりません。
※通信販売ご希望の方は、商品名、機種名、住所、電話番号を明記の上、現金書留にてお申し込み下さい。(送料サービス)
●スタッフ募集！
プログラマー・デザイナーを募集しております。お問い合せください。
資料請求券
Oh!X
8月号
制作：リー・ウェイ・コーポレーション
販売：株式会社ザイン・ソフト
〒676 兵庫県高砂市米田町米田1162-1 TEL.(0794)31-7453

Hope Springs Eternal With My Dear Friends…

ゴールドはシューティング…68K・ミッド・ガルツは、ストーリーをビジュアル・シーンで捕捉しつつフルグラフィック2重スクロール!!
グラフィカル・トラップ!! ステージエンド・ビッグ・モンスター!!…等、シューティング・アクションの基本に回帰します!!
5"2HD 5枚組 ¥12,800(税抜き) 8月11日全国一斉発売!!

セイクリッド・ファンタジー・シリーズ

捏造された真実を見てはならない。
世界の総てを形成したのは何者の意匠によるのか。
時代の流れは、神の意志の集成であるのか。神々こそ偉大であれ。
しかし、人の力をこそ信じたい。そして、この世界に神があることを知れ。
人と世界に依るべき真の姿が見えるか。
壮大なる叙事詩。新たなる時代の叫びを聞け。

GOLD

アークスpro68K (X68000版)
好評発売中! ¥10,000(税抜き)

ゲーマーズ・ホット・アクセス TEL03(5273)4795

※通信販売ご希望の方は、商品名、機種名、住所、氏名、電話番号を明記の上、現金書留にてお申込ください。(送料無料)

株式会社ウルフ・チーム 〒162 東京都新宿区馬場下町61 RK早稲田ビル5F

戦闘体制完了——。

〔ジェノサイド〕

GENOCIDE

X68000 ONLY

7·17発売

大量虐殺前兆（ジェノサイド）
〈リアルバトル&ビジュアルストーリー〉

機能表現120%・迫真のキャラクター&グラフィック

人間を砕くサウンド
〈FM音源＋PCMサウンド〉

¥8,800 ディスク4枚組（税別）

※画面写真は開発中のものです。

2172年————地球（ガイア）存続の運命を担う、超ニューロコンピューターMESIAが稼働した。だが、恐るべき陰謀が潜行していたのだ。その陰謀とは人類大量虐殺。廃墟の都市からエアーズロックの内部へ。ステージごとにパワーを増し巨大化するメシア軍団VS人類の未来をかけて立ち上がった竜ヶ崎健。主人公出生の秘密、メシア反逆の訳——バトルをクリアするたびに数知れぬ謎が明かされていく——。やがてゲームはクライマックスを迎える。プレーヤーの度胆を抜くラストステージ、人類の本当の敵は、一体誰だったのか？GENOCIDE、君の頭脳を直撃する。

ZOOM
COMPUTER SOFT CREATE

◉通信販売ご希望の方は商品名・住所・氏名・電話番号を明記の上、現金書留で当社宛お送り下さい。（送料無料）
◉スタッフ募集！　●プログラマー ●グラフィックデザイナー ●サウンドクリエーター　詳しくは担当：鈴木までご連絡下さい。

株式会社 ズーム　札幌市中央区大通西15丁目 ニューライフ大通弐番館1004　TEL:011·613·0191 FAX:011·613·9570 〒060

GRAPHIC SYNTHESIZER
PSY CLONE
3D RAY TRACING CG TOOL

新発売

サイクロンExpress

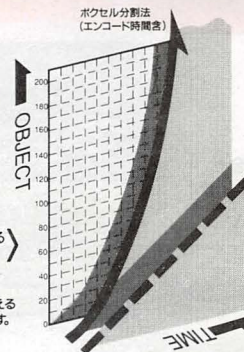
3Dレイ・トレーシングCGツール

高性能モデラー＋リアリズム＋超高速レンダラー

（アンチエリアシング・テクスチャーマッピング・バンプマッピング・属性マッピング＋ボクセル分割）

〈ボクセル分割（空間分割法）による物体数と計算時間〉
—同一条件下における乱反射体—

ボクセル分割法（エンコード時間含）
通常のレイ・トレース法
OBJECT
TIME

※およそ、20個を境にして、物体数が増えるほど、ボクセル分割法が有利になります。

TECHNICAL DATA
Machine……X68000
Pixel………400×400
Object………68
Light………2
Trace-Level…7
Time………20H
（従来タイプ300H）
Design………Pitt Tanaka

ソフトパワーで200倍速……クオリティーは能率に比例します。

サイクロンExpressは物体数が多ければ多いほど威力を発揮する高速レンダリングアルゴリズム…「ボクセル分割」を採用。現状ソフトの2〜900倍（当社比）という高速レンダリングを可能にしました。
又、従来のモデラーのスピードと操作性を飛躍的に向上させました。この結果バンプマッピング、属性マッピング、テクスチャーマッピング及びアンチエリアシングという高度なテクニックを最大限に利用でき、リアリズムの追求に威力を発揮するソフトとして新登場しました。
ペインティングツールとして、Z'sSTAFF（PRO 68K）とデータの互換があります。

サイクロンはアンスのオリジナルCG商品です

●第1回、サイクロンテクニカルセミナーが開催されます。
日時：'89年7月30日(日)10：00AM〜5：00PM　●場所：東京都渋谷区道玄坂2-10-7
会場：新大宗ビルフォーラムエイト　●参加料：3,000円…詳しくはお問い合わせ下さい。

サイクロンExpress

●SHARP版　X68000……………… 78,000円

（注）すでにサイクロン購入ユーザー様で、ユーザー登録ハガキを返送されていない方はお急ぎください。来る7月30日までステップアップサービスキャンペーン実施中です。
詳細はお問い合せください。

サイクロンファミリー 3DレイトレーシングCGツール

■サイクロン68K Ver1.2（SHARP X68000）……………（入門版）58,000円
■サイクロン98LIGHT Ver1.2（NEC PC-9801）…………（入門版）98,000円
■サイクロン98TURBO・EX（NEC PC-9801）…………（業務用）250,000円

★アプリケーション（別売）……サイクロンアニメキット68K Ver1.2…………5,000円
サイクロンアニメキット98 Ver1.2…………5,000円
サイクロンアニメキット68EX…………7,800円
サイクロンアニメキット98EX…………7,800円

株式会社アンス・コンサルタンツ
九州本社/〒810 福岡市中央区平丘町68
phone.092-522-6347　FAX092-521-0400

超低金利クレジットOK

またまた秋葉原でおなじみの

7/15〜8/20

●お近くの方はお
●本体単品で特
●ビジネスソフト定

X68000ACE-HDセット!! 特別ご提供品!!

台数限定　送料¥2,000　※お電話下さい。

ACE-HD ● CZ-611C＋CZ-611D＋M-2HD(10枚)＋ゲーム…定価¥544,800▶ P&A超特価!

12回	¥30,400	24回	¥15,900	36回	¥10,900	48回	¥8,500	60回	¥7,100

ジョイスティック　送料¥500
- X-1PRO 定価¥9,500▶ 特価¥7,800
- ASCII STICK 定価¥6,800▶ 特価¥5,500

P&A超低金利クレジットをご利用ください!!

X68000EXPERT & EXPERT-HD　(送料¥2,000)

シャープ
CZ-8PC2
定価¥69,800
●熱転写プリンター
●48ドット

Ⓐセット：CZ-602C＋CZ-603D＋CZ-8PC2＋M-2HD(10枚)＋ゲーム
…………定価¥510,600▶ P&A超特価(現金価格はお電話下さい)

12回	¥32,800	24回	¥17,200	36回	¥11,800	48回	¥9,100	60回	¥7,600

Ⓑセット：CZ-612C＋CZ-603D＋CZ-8PC2＋M-2HD(10枚)＋ゲーム
…………定価¥620,600▶ P&A超特価(現金価格はお電話下さい)

12回	¥40,000	24回	¥20,900	36回	¥14,400	48回	¥11,200	60回	¥9,300

※モニターをCZ-612D(¥119,800)、CZ-602D(¥99,800)、CU-21CD(¥139,800)に変更の場合も超特価で販売しております。TEL下さい。

※X68000セットでお買い上げの方に
アフターバーナー(定価¥9,200)をプレゼント!!

X68000PRO & PRO-HD　(送料¥2,000)

※プリンターなしの組合せ
※他のプリンターの組合せ
もあります。お電話下さい。

Ⓒセット：CZ-652C＋CZ-603D＋CZ-8PC2＋M-2HD(10枚)＋ゲーム
…………定価¥452,600▶ P&A超特価(現金価格はお電話下さい)

12回	¥28,900	24回	¥15,100	36回	¥10,400	48回	¥8,000	60回	¥6,700

Ⓓセット：CZ-662C＋CZ-603D＋CZ-8PC2＋M-2HD(10枚)＋ゲーム
…………定価¥562,600▶ P&A超特価(現金価格はお電話下さい)

12回	¥36,200	24回	¥18,900	36回	¥13,000	48回	¥10,100	60回	¥8,400

※モニターをCZ-612D(¥119,800)、CZ-602D(¥99,800)、CU-21CD(¥139,800)に変更の場合も超特価で販売しております。TEL下さい。

※X68000セットでお買い上げの方に
アフターバーナー(定価¥9,200)をプレゼント!!

X-1ターボZⅢ

(セットでお買い上げの方にディスケット10枚ジョイカード、ゲーム3種プレゼント)　送料¥2,000

Ⓐセット：X-1ターボZⅢ(CZ-888C＋CZ-860D)＋M-2HD10枚＋ジョイカード＋ゲーム3種……
…. 定価¥269,600▶ 超特価!! TEL下さい。

12回	¥17,200	24回	¥9,000	36回	¥6,200	42回	¥4,800	60回	¥4,000

X-1Gモデル30

台数限定　新品　送料無料

※家庭用TVにつないで2人でゲームを楽しもう!!

- CZ-822C(ブラック)
- AN-58C(RFコンバーター)
- ディスケット10枚
- ゲーム3種
- ジョイカード

P&A超特価¥29,000

中古パソコン　(送料¥2,000)

X68000セット	X68000ACEセット	X-1ターボZ	X-1G/30
●CZ-600C ●CZ-600D	●CZ-601C ●CZ-601D	●CZ-880C ●CZ-880D	●CZ-822C ●CZ-14GB
¥220,000	¥250,000	¥110,000	¥49,000

本体
- CZ-822C ………¥25,000
- CZ-830C ………¥35,000
- CZ-856C ………¥55,000
- CZ-870C ………¥65,000
- CZ-881C ………¥75,000

モニター
- CZ-820D ………¥20,000
- Cu-14GB ………¥15,000

モニター
- Cu-14AG1 ………¥35,000
- Cu-14BD ………¥35,000
- Cu-14AG2 ………¥40,000
- Cu-14H2 ………¥40,000
- CZ-855D ………¥51,000
- MZ-1P17 ………¥25,000
- CZ-8PC2 ………¥35,000
- CZ-8PK6 ………¥42,000

●本広告の掲載の商品の価格については、消費税は含まれておりません。4月1日以降より消費税が付加されますので、ご了承下さい。
● お知らせ 5月21日より営業時間の変更＝平日AM10:00〜PM8:00、日祭AM10:00〜PM7:00

回～60回払いまでOK!!

★頭金なし!★即日発送

P&Aがズバリ超特価セールでご奉仕!!

立寄り下さい。専門係員が説明いたします。
価で受付します。詳しくは電話にてお問合せ下さい。
価の20%引きOK! TELください。

全国通販

超特価でクレジットが組める!!

X68000用ソフトコーナー(送料1ヶ～5ヶまで¥500)

Z's STAFF PRO68K Ver2.0(ツァイト)……定価¥ 58,000➡特価¥ 40,600
C-TRACE68(キャスト)……定価¥ 68,000➡特価¥ 50,300
彩CRONE(アンス・コンサルタンツ)……定価¥ 58,000➡特価¥ 44,600
アニメキット(アンス・コンサルタンツ)……定価¥ 5,000➡特価¥ 4,000
テラッツォ(ハミングバード)……定価¥ 19,800➡特価¥ 15,800
G-68K(OH! BISINESS)……定価¥ 14,800➡特価¥ 11,400
KAMIKAZE(サムシング・グッド)……定価¥ 68,800➡特価¥ 46,800
EW&EI(イースト)……定価¥ 38,800➡特価¥ 28,800
C&Professional Pack(マイクロウェアジャパン)……定価¥ 58,800➡特価¥ 46,000
Final Ver3.2(エーエスピー)……定価¥ 38,000➡特価¥ 30,000
DATA PRO68K CZ220BS……定価¥ 58,000➡P&A特価
CARD PRO68K CZ226BS……定価¥ 29,800➡TEL下さい。!
C compiler PRO68K CZ211LS……定価¥ 39,800➡特価¥ 32,000
OS-9/X68000 CZ219SS……定価¥ 29,800➡P&A特価 TEL下さい。
AI-68K CZ234LS……定価¥188,000➡特価¥143,000
THE福袋V2.0 CZ224LS……定価¥ 9,980➡特価¥ 18,000
SOUND PRO68K……定価¥ 15,800➡特価¥ 12,500
MUSIC PRO68K CZ213MS……定価¥ 15,800➡P&A特価 TEL下さい。
Sampling PRO68K CZ215MS……定価¥ 17,800➡特価¥ 14,000
MUSIC-studio PRO68K 237MS……定価¥ 15,800➡P&A特価 TEL下さい。
MUSIC-PRO68K〔MIDI〕 247MS……定価¥ 18,800➡特価¥ 22,000
New-print Shop 221HS……定価¥ 19,800➡P&A特価
Communication 223CS……定価¥ 19,800➡TEL下さい。!

ゲームソフト(1ヶ～20ヶまで送料¥500)

X68000用
Ⓐ源平討魔伝(電波新聞社)‥定価¥ 7,800➡特価 ¥ 6,200
Ⓑドラゴンスピリット(電波新聞社)定価¥ 8,800➡特価 ¥ 7,000
Ⓒスペースハリアー(電波新聞社)定価¥ 6,800➡特価 ¥ 5,400
Ⓓ熱血高校ドッジボール部(SHARP)‥定価¥ 7,800➡P&A超特価
Ⓔ沙羅曼蛇(SHARP)……定価¥ 8,800➡P&A超特価
Ⓕフルスロットル(SHARP)…定価¥ 8,800➡P&A超特価
Ⓖ琥珀色の遺言(リバーヒルソフト)・定価¥ 9,800➡特価 ¥ 7,800
Ⓗザ・スーパーラスベガス(日本デグスタ)…定価¥12,800➡特価 ¥10,200
Ⓘマイト・アンド・マジック(スタークラフト)定価¥ 9,800➡特価 ¥ 7,800
Ⓙザ・リターン・オブ・イシター(SPS) 定価¥ 7,800➡特価 ¥ 6,200
Ⓚ信長の野望(全国版)(KOEI)‥定価¥ 9,800➡特価 ¥ 7,800
Ⓛ麻雀悟空(シャノアール)…定価¥ 7,800➡特価 ¥ 6,200
ⓂマーダークラブDX(リバーヒルソフト)定価¥ 7,800➡特価 ¥ 6,200
Ⓝザキングオブシカゴ(ボーステック)…定価¥12,800➡特価 ¥10,200
Ⓞ今夜も朝までパワフルまあじゃん2(dB-SOFT)……定価¥ 7,800➡特価 ¥ 6,200
Ⓟ三国志(光栄)……定価¥14,800➡特価 ¥12,000

モデムコーナー (送料¥1,000)

ⒶMD-2400B(オムロン)……定価¥49,800➡特価¥36,000
ⒷMD-2400F(オムロン)……定価¥59,800➡特価¥42,000
ⒸPV-A2400MNP4(アイワ)……定価¥46,800➡特価¥35,000
ⒹPV-A24MNP5(アイワ)……定価¥54,800➡特価¥41,000

カラービデオプリンター (送料¥1,000)

Ⓐセット:CZ-6PVI……定価¥198,000➡TEL下さい

12回	13,400	24回	7,000	36回	4,800	48回	3,700

P & A 特選パソコンラック (送料無料) 移動自由(キャスター付)

カラーイメージスキャナ (送料¥1,000)⊕ジェット

ⒶCZ-8NSI…定価¥188,000➡P&A超特価!!
ⒷJX-100…定価¥ 89,800➡P&A超特価!!
ⒸJX-200…定価¥198,000➡
Ⓓ10-730…定価¥230,000➡P&A超特価!!
Ⓔ10-735…定価¥248,000➡P&A超特価!!

プリンター

ⒶCZ-8PC3……定価¥ 65,800▶P&A超特価
ⒷCZ-8PC4……定価¥ 99,800▶P&A超特価
ⒸCZ-8PK8……定価¥152,000▶P&A超特価
ⒹCZ-8PK9……定価¥ 89,800▶P&A超特価
ⒺCZ-8PK6……定価¥159,000▶特価¥69,000

周辺機器コーナー(送料¥1,000)●その他の周辺機器はお電話下さい。

ⒶCZ-8BSI(FM音源ボード)……定価¥23,800➡P&A超特価
ⒷCZ-8RLI(データレコーダ)……定価¥24,800➡特価¥20,000
ⒸCZ-6BE1A(1M RAM)……定価¥38,000➡P&A超特価
ⒹCZ-6BE4(4M RAM)……定価¥138,000 ➡P&A超特価
ⒺCZ-6BP1(数値演算)……定価¥79,800➡特価¥61,000
ⒻCZ-6VTI(カラーイメージユニット)……定価¥69,800➡P&A超特価
ⒼCZ-6EBI(拡張I/Oボックス)……定価¥88,000➡特価¥69,000
ⒽAN-160SP(アンプ内蔵スピーカーシステム)‥定価¥59,800➡特価¥47,000

中古パソコンはP&Aにおまかせ!!

その場で高価現金買取り・高価下取りOK!!

■まずはお電話下さい。
03-651-1884
FAX:03-651-0141

■下取り・買取りでお急ぎの方、直接当社に来店、または、宅急便にてお送り下さい。

●下取りの場合……価格は常に変動していますので査定額をお電話で確認して下さい。(差額は、P&A超低金利クレジットをご利用下さい。)

●買取りの場合……現品が着き次第、2日以内に買取り金額を連絡し、振込み、又は書留でお送り致します。

●近郊の方は、P&A本店まで、直接お持ち下さい。
即金にて、¥1,000,000までお支払い致します。

通信販売お申し込みのご案内

〔現金一括でお申し込みの方〕
●商品名およびお客様の住所・氏名・電話番号をご記入の上、代金を当社まで、現金書留でお送りください。(プリンター・フロッピーの場合、本体使用機種名を明記のこと)
〔銀行振込でお申し込みの方〕
●銀行振込ご希望の方は必ずお振込みの前にお電話にてお客様のご住所・お名前・商品名等をお知らせください。
(電信扱いでお振込み下さい。)

〔振込先〕住友銀行 新小岩支店
当No.263914 ㈱ピー・アンド・エー

〔クレジットでお申し込みの方〕
●電話にてお申し込みください。クレジット申し込み用紙をお送りいたしますので、ご記入の上、当社までお送りください。
●現金特別価格でクレジットが利用できます。残金のみに金利がかかります。
●1回～60回払いまで出来ます。但し、1回のお支払い額は3,000円以上。

アフターサービス万全
全商品保証付。専門の担当者がお客様の立場で対応します。
初期不良、輸送トラブル etc.
万が一初期不良、輸送トラブルが発生しました際には、即交換させていただきます。

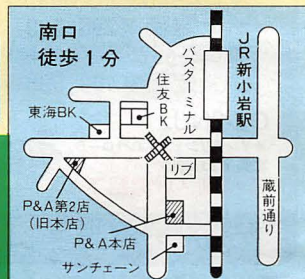
超低金利クレジット率

回数	1	3	6	10	12	15	18	24	36	48	60
利率(%)	1.5	2.0	3.0	4.5	4.5	7.5	9.0	9.5	13	17	22

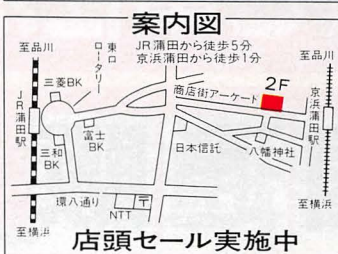
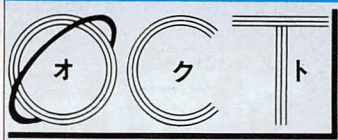
●定休日/毎週水曜日=第3水曜・木曜は連休とさせていただきます(祭日の場合は翌日になります)

●マイコン
●ビデオ
●ビデオテープ

P&A
株式会社ピー・アンド・エー
〒124 東京都葛飾区新小岩2丁目1番地19号
☎03-651-0148(代) FAX.03-651-0141

営業時間
平日AM10:00～PM8:00
日祭AM10:00～PM8:00

●現金書留及び銀行振込でお申し込みの方は、上記商品の料金に3%加算の上でお申し込み下さい。詳しくは、お電話でお問い合せ下さい。

パソコンプラザ

オクト OCT

―オクトで始まるパソコンワールド―

☎03-730-6271

●営業時間 AM 11:00～9:00/日曜・祭日PM7:00 電話一本で、ハイ即納

〒144 東京都大田区蒲田4-6-7 FAX 03-730-6273

●定休日毎週火曜日 祭日の場合翌日になります。

オクト ラクラククレジット

12回	4.5%	24回	10%	36回	14%	48回	18%

案内図

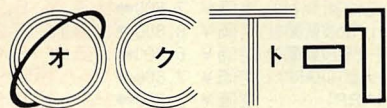
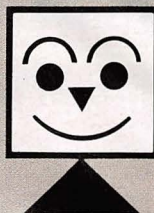
店頭セール実施中

OCT-1 システム インフォメーション

- ▶全商品保証付(メーカー保証)
- ▶超低金利ハッピークレジット(1回～60回)頭金ナシOK!
- ▶ボーナス一括払いOK! ボーナス2回払いOK!!
- ▶配達日の指定OK!(万全なサポート体制)
- ▶商品の組合せ自由! オクトフリーダムシステム
- ▶店頭デモンストレーション実施中

オクト セレクテッドシステム

広告掲載商品以外の製品も取扱っております。

オクト-1 蒲田

夏って好きだナ!! だから思いっ切り遊んじゃおっと!!

X68000フェア開催中!!

OPEN

《新製品発売記念プレゼント実施中》★セットでお買い上げの方には、アフターバーナー(¥9,200)をプレゼントいたします。

お好みのセットをお選び下さい。

送料無料

●3Mバイトの大容量メモリ
●40Mバイトハードディスク搭載

EXPERT・EXPERT-HD

●CZ-602C(BK) 定価¥356,000
●CZ-612C(BK) 定価¥466,000

現金特価!! 推選 お電話下さい。

●拡張I/Oポート4スロット装備
●2Mバイトの大容量メモリ

PRO・PRO-HD

●CZ-652C(GY/BK) 定価¥298,000
●CZ-662C(GY/BK) 定価¥408,000

CZ-8NJ2
●インテリジェントコントローラ
定価¥23,800
超特価!お電話下さい。

15型カラーディスプレイTV

CZ-612D-GY/BK NEW
定価¥119,800

15型カラーディスプレイTV

CZ-602D-GY/BK NEW
定価¥ 99,800

14型カラーディスプレー

CZ-603D-GY/BK
定価¥84,800

21型カラーディスプレイ

CU-21CD
定価¥139,800

Ⓐ CZ-602C＋CZ-612D＋MD-2HD10枚＋ゲーム ………定価¥475,000▶ウフフ。お買徳ですヨ!
Ⓑ CZ-612C＋CZ-612D＋MD-2HD10枚＋ゲーム ………定価¥585,800▶超低金利クレジットをご利用下さい。
Ⓒ CZ-652C＋CZ-612D＋MD-2HD10枚＋ゲーム ………定価¥417,800▶電話一本。ハイ即納。
Ⓓ CZ-662C＋CZ-612D＋MD-2HD10枚＋ゲーム ………定価¥527,800▶超特価!電話下さい。

Ⓔ CZ-602C＋CZ-602D＋MD-2HD10枚＋ゲーム ………定価¥455,800▶超特価!電話下さい。
Ⓕ CZ-612C＋CZ-602D＋MD-2HD10枚＋ゲーム ………定価¥568,800▶ウフフ。お買徳ですヨ!
Ⓖ CZ-652C＋CZ-602D＋MD-2HD10枚＋ゲーム ………定価¥397,800▶超低金利クレジットをご利用下さい。
Ⓗ CZ-662C＋CZ-602D＋MD-2HD10枚＋ゲーム ………定価¥507,800▶電話一本。ハイ即納。

Ⓘ CZ-602C＋CZ-603D＋MD-2HD10枚＋ゲーム ………定価¥440,800▶電話一本。ハイ即納。
Ⓙ CZ-612C＋CZ-603D＋MD-2HD10枚＋ゲーム ………定価¥550,800▶超特価!電話下さい。
Ⓚ CZ-652C＋CZ-603D＋MD-2HD10枚＋ゲーム ………定価¥382,800▶ウフフ。お買徳ですヨ!
Ⓛ CZ-662C＋CZ-603D＋MD-2HD10枚＋ゲーム ………定価¥492,800▶超低金利クレジットをご利用下さい。

Ⓜ CZ-602C＋CU-21CD＋MD-2HD10枚＋ゲーム ………定価¥495,800▶超低金利クレジットをご利用下さい。
Ⓝ CZ-612C＋CU-21CD＋MD-2HD10枚＋ゲーム ………定価¥605,800▶電話一本。ハイ即納。
Ⓞ CZ-652C＋CU-21CD＋MD-2HD10枚＋ゲーム ………定価¥437,800▶超特価!電話下さい。
Ⓟ CZ-662C＋CU-21CD＋MD-2HD10枚＋ゲーム ………定価¥547,800▶ウフフ。お買徳ですヨ!

オクトで始まるパソコンワールド。店頭にて、ゲームソフト25%OFF!!(税別)超低金利ハッピークレジットをご利用ください。

※クレジットの回数は1回～60回、ボーナス併用などありますのでお電話でお問合せ下さい。

■本体セット:送料¥2,000 ●店頭デモ実施中…専門の係員が詳細にアドバイス致します。ぜひご来店下さい。

※上記料金には、消費税は含まれておりません。消費税が付加されますので、詳しくは、電話でお問合せ下さい。

■特に人気のある商品によっては、しばらくお待ち願うことがありますのでご了承下さい。

厳選された製品を、より安く、より早く、皆様のお手元に!!

広告掲載商品以外の
製品も取扱っております。

ラストチャンス！ X68000ACE-HD超特価セール!!

※セットでお買上げの方にはアフターバーナー(ゲーム)をプレゼント!!

限定 送料￥2,000

推奨セット

Ⓐ CZ-611C＋CZ-603D＋MD-2HD＋ゲーム
……▶超特価／TEL下さい。

Ⓑ CZ-611C＋CZ-602D＋MD-2HD＋ゲーム
……▶超特価／TEL下さい。

Ⓒ CZ-611C＋CZ-611D＋MD-2HD＋ゲーム
……▶超特価／TEL下さい。

Ⓓ CZ-611C＋Cu-21CD＋MD-2HD＋ゲーム
……▶超特価／TEL下さい。

㊙超特価
絶対！
お徳デス!!

X68000 ACE-HD

※超低金利クレジットご利用下さい。1回～60回払い、頭金ナシ／ボーナス1回払い、ボーナス2回払いOK／

なんとウレシイ／X-1G超特価（送料無料）

限定

X-1G(本体)
- CZ-882C
- MD-2HD10枚
- ジョイカード（連射）
- ゲームソフト1本

買わなきゃソンをする!!
早い者勝ち!!

㊢大特価￥29,000

型名	商品	特価	特価	型名	商品	定価	特価
CZ-6BE1	1MB増設RAMボード	￥38,000	大特価	CZ-6EB2	拡張I/Oボックス	￥88,000	大特価
CZ-6BE2	2MB増設RAMボード	￥79,000	大特価	CZ-8TMZ	モデムユニット	￥49,800	大特価
CZ-6BG1	GP-IBボード	￥59,800	大特価	CZ-6BN1	スキャナ用パラレルボード	￥29,800	大特価
CZ-6BP1	プロセッサ・ボード	￥79,800	大特価	CZ-8NT1	トラックボール	￥13,800	大特価
CZ-6BC1	FAXボード	￥79,800	大特価	CZ-6BU1	ユニバーサルI/Oボード	￥39,800	大特価
CZ-6BM1	MIDボード	￥26,800	大特価	AN-160SP	アンプ内蔵スピーカ	￥59,800	大特価
AN-8TV	パソコンチューナー	￥35,800	大特価	CZ-6PV1	カラービデオプリンタ	￥198,000	大特価
CZ-8NS1	カラーイメージスキャナ	￥188,000	大特価	CZ-6VT1-BK	カラーイメージユニット	￥69,800	大特価

熱転写カラー漢字プリンター 用紙プレゼント 送料無料

CZ-8PC4 ￥99,800
- 48ドットサーマルヘッド
- B5～B4まで
- ハガキ可能
- カラー対応

大特価 オクト推選 TEL下さい／

①CZ-8PK7（24ピン80桁）
定価￥122,000……大特価・TEL下さい。

②CZ-8PK8（24ピン136桁）
定価￥152,000……大特価・TEL下さい。

③CZ-8PK9
定価￥89,800……大特価・TEL下さい。

④CZ-8PC3（24ドット漢字カラー）
定価￥65,800……大特価・TEL下さい。

パソコンラック 推奨 送料無料

5段キャスター付
キーボード が収納できるから、手元でマウス操作がラクラクできる
棚板5段のマルチに活用できるディスク。
ウーン、こいつはデキル！
1325(H)×640(W)×700(D)
特価￥16,000

4段キャスター付
どんなパソコンにもフレキシブルに対応！
使い易いデスクです。
1245(H)×614(W)×600(D)
特価￥12,000

X68000ソフト大セール実施中 ※ゲームソフトオール25%off

〈グラフィック〉 ●Z's STAFF PRO68K Ver.2.0
（シャフト）定価￥58,000
オクト特価￥41,000

〈データベース〉 ●KAMIKAZE
（サムシンググッド）￥定価68,000
オクト特価￥47,000

〈グラフィック〉 ●C-TRACE68
（キャスト）定価￥68,000
オクト特価￥51,000

〈C言語〉 ●C&Professional Pack
（マイクロウェア ジャパン）定価￥58,000
オクト特価￥44,800

型名	商品	定価	特価
BUSINESS PRO68K	統合型表計算	￥68,000	大特価
CARD PRO68K	カード型データベース	￥29,800	大特価
DATA PRO68K	コマンド型データベース	￥58,000	大特価
COMMUNICATION PRO68K	通信ソフト	￥19,800	大特価
OS-9 X68000	マルチタイム リアルタイム オペレーティング システム	￥29,800	大特価
MUSIC PRO68K	楽譜ワープロ	￥18,800	大特価
SOUND PRO68K	サウンドエディタ	￥15,800	大特価
NEW PRINT SHOP PRO68K	ポップアートツール	￥19,800	大特価
C-COMPILER PRO68K	Cコンパイラ	￥39,800	大特価
EW	ワープロ	￥38,000	￥29,800
G-68	グラフィックツール	￥14,800	￥12,000
E-68K	スプライトエディタ	￥19,800	￥16,000

店頭ゲームソフトオール25%off／ビジネスソフト 25%より特価中

●尚、送料として1ケ￥500、2ケ￥700、3ケ以上で￥1,000となります。

★通信販売お申込みのご案内★ 〒144 東京都大田区蒲田4-6-7 TEL：03-730-6271

お申込みはお電話でお願いします。お客様の〈住所〉〈氏名〉〈電話番号〉及び〈商品名〉をお知らせ下さい。●入金確認後ただちに商品をご送付いたします。

現金一括払い
銀行振込：お近くの銀行より（電信扱い）にてお振込み下さい。
現金書留：封筒の中に住所・氏名・商品名をご記入の上当社までお送り下さい。

クレジット
専用お申込用紙をお送り致します。ので、必要事項をご記入、ご捺印の上ご返送下さい。手続きは簡単です。

低金利クレジットをご利用下さい
オクトラクラククレジット

12回	4.5%	24回	10%	36回	14%	48回	18%

振込先
富士銀行 久ヶ原支店 ㊵No.1824
三菱銀行 蒲田支店 ㊵No.0278691
株式会社 億人（オクト）

※掲載の価格は6/20現在ですので、まずは、お電話にてご確認ください。8/22・23は連休とさせていただきます。
※上記料金には、消費税は含まれておりません。消費税が付加されますので、詳しくは電話でお問合せ下さい。
※銀行振込、または、現金書留でご注文の際には、あらかじめ電話でご確認の上、お申し込み下さい。

オクトで始まるパソコンワールド。
店頭にて、ゲームソフト25%OFF!!（税別）超低金利ハッピークレジットをご利用ください。

ツクモ おもいっきり！夏フェア

ツクモの夏はあ・つ・い

2万円以上お買い上げの場合送料無料！

7/15～8/1の期間中、素敵な事が‥‥

★店頭にて1万円以上お買い上げのお客様には、カラクジなしのクジ引きで素敵な景品をプレゼントしまぁ～す♡

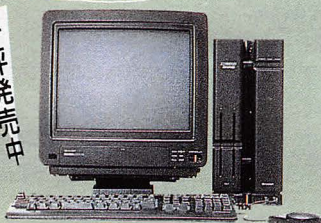
お申し込みの☎は03-251-9911へ！ 夜10時まで受け付けております。
代金引換え配達、月々¥3,000のクレジット、夏のボーナス一括払いなどご希望に応じてお支払いはらくらく！

X68000シリーズ好評発売中！

「X68000オリジナルグッズコーナー」も増えて更に人気上昇中！

アクセサリーいろいろ

ツクモオリジナルキーボード延長ケーブル ツクモ特価¥1,980
キーボードシリコンカバー 標準価格¥2,800
キーボードセフティカバーASC108 標準価格¥2,400
キーボードダストカバーADC108 標準価格¥1,000

好評発売中

X68000 EXPERT / EXPERT HD
PERSONAL WORKSTATION
本体＋キーボード＋マウス・トラックボール
CZ-602C-BK(ブラック)・GY(グレー) 定価¥356,000
HDタイプ CZ-612C-BK(ブラック) 定価¥466,000

好評発売中

X68000 PRO / PRO HD
PERSONAL WORKSTATION
本体＋キーボード＋マウス
CZ-652C-GY(グレー)・BK(ブラック) 定価¥298,000
HDタイプ CZ-662-GY(グレー)・BK(ブラック) 定価¥408,000

TSUKUMO

ディスプレイ

CZ-602D ドットピッチ0.39㎜タイプ‥‥‥定価¥99,800
CZ-612D ドットピッチ0.31㎜タイプ‥‥‥定価¥119,800
CZ-603D ドットピッチ0.31㎜タイプ‥‥‥定価¥84,800
CU-21CD 21インチディスプレイ‥‥‥定価¥139,800
■オプション
CZ-6ST1 (チルト台)‥‥‥定価¥5,800
CZ-6TU (RBGシステムチューナー)‥‥‥定価¥33,100
BF-68PRO (高性能CRTフィルター)‥‥‥定価¥19,800

周辺機器

CZ-6BE1 1MB内蔵RAM(CZ-600C専用)‥‥‥定価¥35,000
CZ-6BE1A 1MB内蔵RAM(ACE・PROシリーズ専用)‥‥‥定価¥38,000
CZ-6BE2 2MB増設RAMボード‥‥‥定価¥79,800
CZ-6BE4 4MB増設RAMボード‥‥‥定価¥138,000
CZ-6BC1 FAXボード‥‥‥定価¥79,800
CZ-6BP1 数値演算プロセッサボード‥‥‥定価¥79,800
CZ-6BM1 MIDIボード‥‥‥定価¥26,800
CZ-6BG1 GP-IBボード‥‥‥定価¥59,800
CZ-6BU1 ユニバーサルI/Oボード‥‥‥定価¥39,800
CZ-6BF1 拡張RS-232Cボード‥‥‥定価¥49,800
CZ-6VT1 カラーイメージユニット‥‥‥定価¥69,800
CZ-8NS1 カラーイメージスキャナ‥‥‥定価¥188,000
AN-S100 アンプ内蔵スピーカーシステム(2本1組)‥‥‥定価¥55,300
※待望のインテリジョントコントローラー登場！
これであなたの部屋はゲームセンター‥‥。
CZ-8NJ2‥‥‥好評販売中！

お勧めソフトウェア

Kamikaze(神風) 総合型スプレッドシート‥‥‥ツクモ特価¥57,800
SOUND PRO-68K サウンドエディタ‥‥‥定価¥15,800
MUSIC PRO-68K ミュージックツール‥‥‥定価¥18,800
Sampling PRO-68K AD PCM活用ソフト‥‥‥定価¥17,800
Musicstudio PRO-68K V1.1 MIDIマルチレコーディングソフト‥‥‥定価¥28,800
MUSIC PRO-68K(MIDI) MUSIC PRO-68KのMIDI版 定価¥28,800
ソングラブラリ〈101曲集〉 MUSIC PRO-68Kデータ曲集 定価¥ 8,800
Communication PRO-68K 通信ソフト‥‥‥定価¥19,800
た～みのる 通信ソフト‥‥‥ツクモ特価¥10,900
DATA PRO-68K リレーショナルデータベース‥‥‥定価¥58,000
CARD PRO-68K カード型データベース‥‥‥定価¥29,800
システム手帳リフィル集 CARD PRO-68K用フォーム集‥定価¥9,800
活用フォーム集 CARD PRO-68K用フォーム集‥‥‥定価¥9,800
Z's STAFF PRO-68K Ver2.0 グラフィックツール‥‥‥ツクモ特価¥49,300
New Print Shop PRO-68K 高機能ポップアートツール‥‥‥定価¥19,800
サイクロン Express 2.0 レイトレーシングソフトウェア‥‥‥ツクモ特価¥67,000
アニメキット(サイクロン68Kが必要) レイトレ・アニメーションツール‥‥‥ツクモ特価¥4,200
C-TRACE 68 レイトレーシングソフトウェア‥‥‥ツクモ特価¥57,800
C COMPILER PRO-68K C言語開発セット‥‥‥定価¥39,800
Final X68000 マルチファイル・スクリーン・エディタ ツクモ特価¥32,300
AI-68K AIプログラム開発ツール‥‥‥定価¥188,000
REDUCE 数式処理用ソフト‥‥‥ツクモ特価¥195,000
OS-9/X68000 X68000用OS-9‥‥‥定価¥29,800
C & プロフェッショナルパッケージ OS-9/X68000用Cコンパイラセット‥‥‥定価¥58,000
mFORTH Compiler FORTHコンパイラセット ツクモ特価¥18,800
Human68K Ver2.0 Human68KのNEWバージョン‥‥‥定価¥9,800
※その他、ゲームソフトも続々発売中ですので、詳しくはお尋ね下さい。

今、大容量のハードディスクが大人気！

●アイテックハードディスク
IT X-203(20MB 28ms)
ツクモ特価¥69,800
消費税別途¥2,094
IT X-403(40MB 29ms)
ツクモ特価¥99,800
消費税別途¥2,994

X-203/403はブラックかグレーかをご指定下さい。

X1 turbo Z III セット
●CZ-888C-BK‥‥‥¥169,800
●CZ-860D-BK‥‥‥¥92,200
ツクモ特価販売中！

NEW MIDIセット

Aセット

MT-32 MIDI音源‥‥‥定価¥64,000
CZ-6BM1 MIDIボード‥‥‥定価¥26,800
CZ-247MS MUSIC PRO-68(MIDI)‥‥‥定価¥28,800
ツクモ特価¥98,000
消費税別途¥2,940

Bセット

MT-32 MIDI音源‥‥‥定価¥64,000
CZ-6BM1 MIDIボード‥‥‥定価¥26,800
CZ-252MS Musicstudio PRO-68K V1.1‥‥‥定価¥28,800
ツクモ特価¥98,000
消費税別途¥2,940

※Music studioデータ曲集も発売中！ 各¥5,800
SF-001 国本佳宏／知恵ある暮しの味
SF-002 佐久間正英／インセクト
SF-003 本多俊之／ピーセス・オブ・ワーク
SF-004 戸田誠司／あの娘のDNA

モデム

オムロン MD12FS(300/1200ボー)‥‥‥ツクモ特価¥17,800
アイワ PV-A1200MK3(300/1200ボー)‥‥‥ツクモ特価¥16,800
アイワ PV-A24MNP5(300/1200/2400ボー)MNP5‥‥‥ツクモ特価¥46,600

やっぱり、カラープリンターが欲しくなる

●カラー漢字24ドット熱転写プリンター
CZ-8PC3‥‥‥定価¥65,800
●カラー漢字48ドット熱転写プリンター
CZ-8PC4‥‥‥定価¥99,800
●カラーイメージジェットプリンター
IO-730‥‥‥定価¥230,000

電子手帳&ポケコンもツクモで‥

シャープ PC-E200
定価¥22,000
特価¥17,800

シャープ PA-8500
定価¥28,000
特価¥24,800
大型4行表示、データスケジュール管理に便利、ICカード、プリンタで更に発展するハイグレードタイプ

シャープ PC-E500
定価¥28,800
特価¥24,800

営AM10時～PM7時 休7/27を除く毎週木曜日

ツクモ 7号店 ☎03-253-4199
通信販売部 ☎03-251-9911
■ツクモ5号店 ☎03-251-0531
■ニューセンター店 ☎03-251-0987
■名古屋1号店 ☎052-263-1655
■名古屋2号店 ☎052-251-3399
■ツクモ札幌 ☎011-241-2299

ツクモは「スーパーX PRO SHOP」です。

PRO STAFF ツクモ
九十九電機㈱ 〒101-91 東京都千代田区神田郵便局私書箱135号

全国代金引き換え配達
お申し込みは☎03-251-9911へお電話1本！
商品到着の際、玄関でお会計ができます。配達日の指定もできます。

夏・冬、ボーナス2回払い受付中
月々¥3,000以上の均等払いも頭金なし。

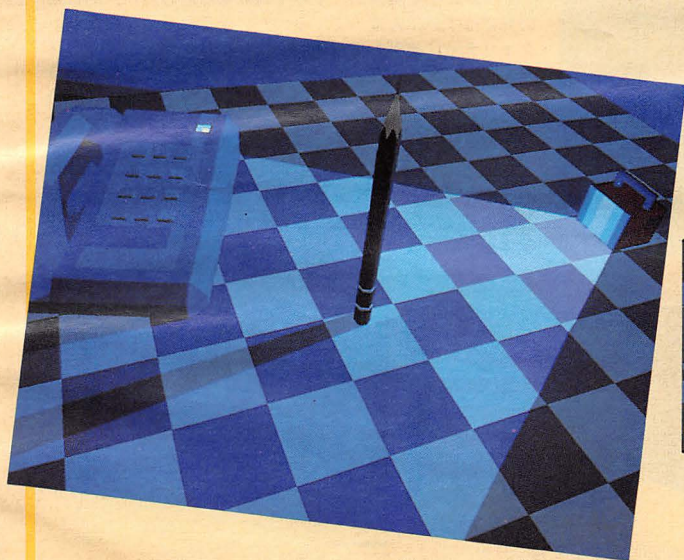
現金書留なら
〒101-91 東京都千代田区神田郵便局私書箱135号
九十九電機㈱通信販売部

銀行振込なら
事前に☎でお届け先をご連絡下さい。
富士銀行 神田支店(普)No.894047

★表示価格には消費税は含まれておりません。

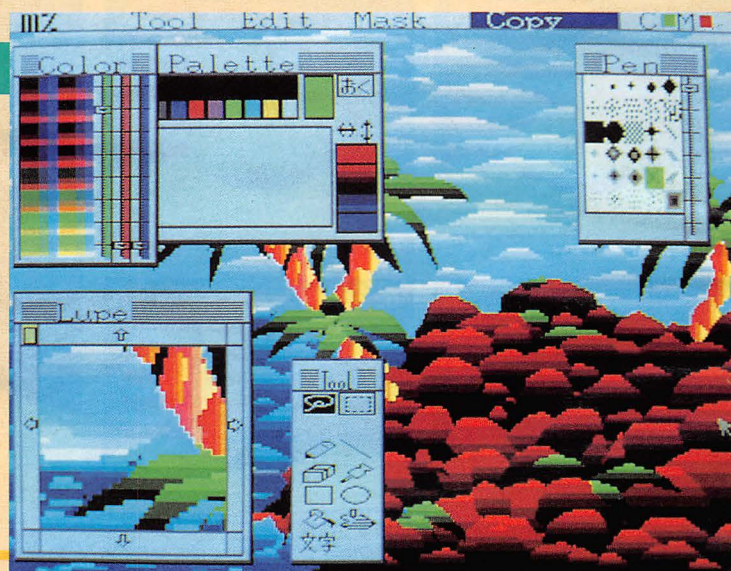
Oh!X Graphic Gallery

DōGA・CGアニメーション講座

まず今月の「寺田の教育的指導」では，寺田氏自身がサンプルとしてヘリコプターのアニメーションを用意して，自らを指導しようということにあいなりました。まずは，ごゆっくりとご鑑賞のうえ，本文ページをお読みください。

こちらは，かまたゆたか氏がつくった光環境のシミュレーションです。CGAシステムの隠れ機能を使っているそうで，かまた氏によると「非常に高度な（姑息な）テクニックを使用しているので，皆さんはまねしちゃだめよ」とのことです。

MZ-2500グラフィックエディタ作成講座

最近MZ-2500ユーザーとなった山田純二くんに，画餅AMA-25hのグラデーション機能を試してみてよ。と言ったらあっというまに作ってくれたのがこのサンプル。今月登場した，パレット，ルーペなどのウィンドウにも注目してください。

特集2

「表示」から「描写」へ，「計算」から「制作」へ

3Dグラフィックの深淵へ

まず基本原理の理解，そして応用。前回は3D計算が中心だったが，今回はレンダリング，モデリングについて補足する。応用編とはいっても，これでやっと3Dグラフィック作品作成への糸口をつかんだにすぎない。本当の深淵はここから先に広がっているのだから。

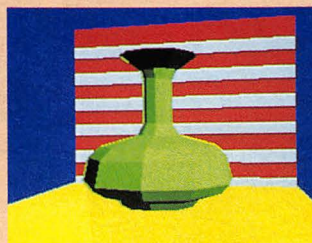
Zバッファによる質感表現

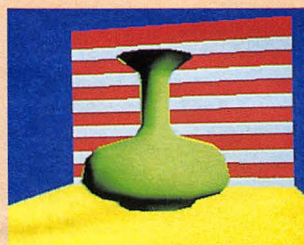
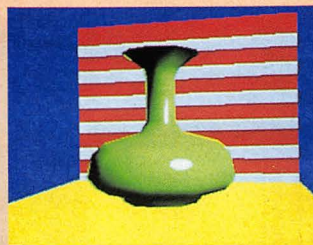
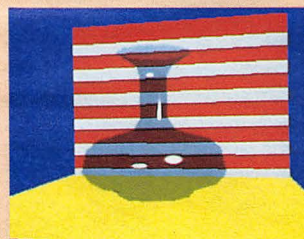
先月行ったZバッファアルゴリズムによる描画ルーチンにスムースシェイディング，透明体の処理をつけ加えた。スムースシェイディングのアルゴリズムは各頂点の色を線的に補間するグローシェイディング，法線を補間して膨らみをシミュレートするフォンシェイディングを用いている。

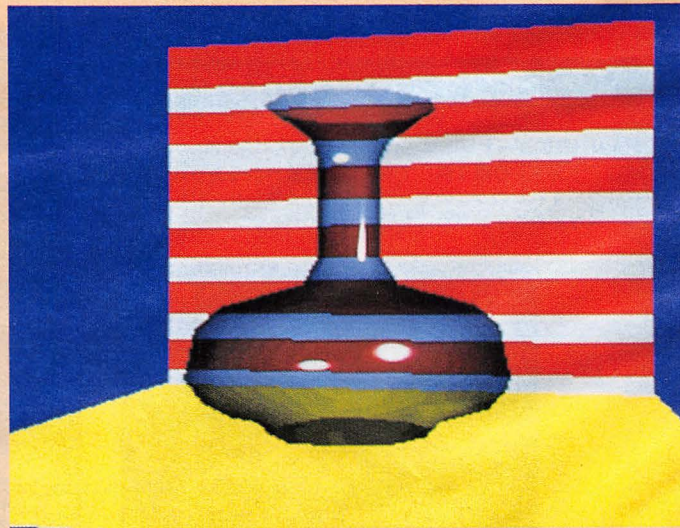
写真は左からコンスタントシェイディング（スムースシェイディングなし），グローシェイディング，フォンシェイディングによる描画例。すべて輪郭線は同じだが，見た目の丸さはずいぶん違ってくるのがわかるだろう。ただし，クオリティに比例して描画時間はかなりかかってくる。フォンシェイディングで透明体などを処理すると，先月のコンスタントシェイディングのみのプログラムに比べて10倍以上の時間がかかる。

下は透明体の例。正確な透明体をシミュレートしているわけではないので，屈折などはできないが，擬似的に光線の減衰を加えることでかなりそれらしく見えるようになる。従来の散乱反射体の指定と組み合わせることで色つき半透明体なども作成できるようになった。

質感表現の応用例。影ができないのでイマイチ浮いて見えるが，このあたりがZバッファの限界だろう。へたに影の生成を加えると（不可能ではない）レイトレーシング並みの時間がかかってしまいそうだ。手作業で影のオブジェクトを加えてやるのがいちばんだ。

なお，スムースシェイディングに必要な頂点ごとの法線はデータコンバート後，CGAのモデリングツールを使って自動生成した。面が多くなると結構手間がかかるが，それでも手作業で行うよりはずっと楽ができる。

コンスタントシェイディング

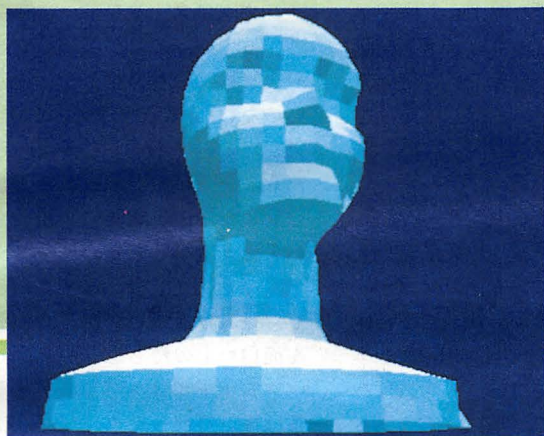
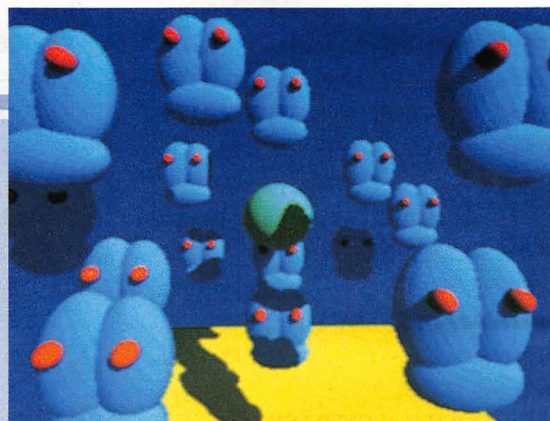
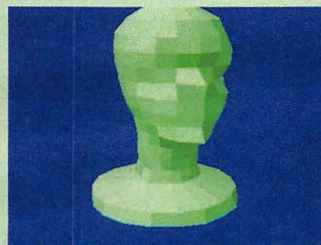
同じく透明体

散乱反射体

散乱反射体＋ハイライト

透明体

輪郭を2度描き

ビデオカメラを使った3Dデジタイズ

できるだけ簡単なシステムで立体取り込みを実現する方法を考えてみた。実際に右のような物体をいろいろな分解能で取り込んで描画したのが以下の写真だ。

右端の大きな写真は約2000個のポリゴンを入力し，その一部を切り取ったもの。これくらいポリゴンが増えると，描画時間（1分強）よりもファイルの読み込みやデータセットの時間のほうがかかってしまうようだ。めいっぱいメモリを使えば標準メモリでも2000～3000個のポリゴンが扱えるので，その気になればかなり本格的なCG制作もできそうだ。なお，ここで作ったデータは簡単にCGAフォーマットに変換できる。

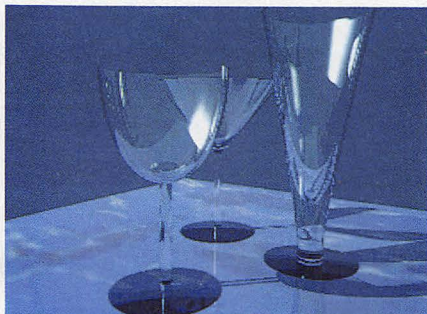
スムースシェイディングの実際。これらの方法では頂点ごとに仮想法線ベクトル（その頂点の向き）を必要とするので，隣合った面の向いている方向の平均をとって指定する。

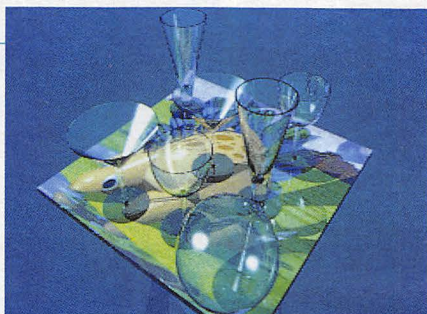
スキャンラインZバッファアルゴリズムでスムースシェイディングをかける場合には，アルゴリズムの都合上，立体を構成するポリゴンに凹多角形を使うと誤動作する（滑らかに補間されない）ので凹多角形は使わないようにしましょう（右の写真を参照のこと）。

先月も紹介したアンス・コンサルタンツのレイトレーシングツール，サイクロンExpress。VOXEL分割の威力はやはり凄い。

写真はどれもオブジェクト数50個程度を使用しているが演算時間はすべて数時間ですむ。いちばん下の鏡面体を使用したものでも，数値演算プロセッサつきで2時間12分で終了する（256×256ドット，トレースレベル4）。

新製品C-TRACE68 TP2

高速ボードはメタボールを目指す

キャスト ☎03(705)0656

このグラスが45分で描ける

反射・屈折がいっぱいのデラックス版

高速レイトレエンジンT-800

もうお馴染みのC-TRACEに超高速バージョンが登場した。それも，ハードウェア方面からの大幅なチューンナップである。もうご存じかと思うが，本誌でもすでに紹介があったトランスピュータボードである。これを試用する機会に恵まれたので，取り急ぎレポートする。

トランスピュータボードは高速32ビットプロセッサT-800（20MHz）で演算を，表示をX68000で行うレイトレーシング専用のハードウェアだ。仮にも16ビットマシンに積むボードで，MFLOPS（メガフロップス：1秒間に何100万回浮動小数点演算を行えるか）などという単位が出現するとは思ってもみなかった。トランスピュータボードのドライバには汎用性がないので，数値演算プロセッサのように一般のアプリケーションが恩恵をこうむるわけではないのが惜しいところだが，ちゃんとドライブしたときのスピードといったら，それを補って余りあるものである。

試しに，3月号でC-TRACEのバージョン2.0をレビューしたときに使ったサンプル（グラスが3つ並んでたあれ）を描画させてみた。バージョン2.0では，数値演算プロセッサつきでも10時間余りかかったが……45分。実に1桁も速度が違う。たった1桁と思うなかれ。今までは寝る前にセットしておいて起きる頃にはできているのがパターンだったが，これからは漫画でも読んでいる間に終わってしまうのだ。さらにドットを粗く取るテスト描画のときなど，見る見るうちに終わってしまう。

高速化は，なにを意味するのか。浮いた計算時間が，シーンを充実させるほうに回せるのである。オブジェクトの数を増やすもよし，マッピングに凝るもよし，構図をいろいろ変えるもよし，オブジェクトの配置を気のすむまで検討するもよし。可能性を広げるおいしいボード。ただし値段も凄いが。

レイトレ新次元！　メタボール登場？

トランスピュータボードについてきたサンプルのディスクを見ていると，「メタボール」の文字が入ったラベル。おお，これはあの，自由曲面の素，憧れのメタボールか？

僕自身，メタボールについての知識はあったが，扱うのは初めてだ(嬉しい！)。まだ完成バージョンではないようだったが，データのフォーマットを手探りで調べつつ，サンプルを作った。

メタボールはかなり変わった手法。これまでのレイトレーシングでは，プリミティブに使えたのが2次曲面と平面くらいだったので，積木のように組み合わせて作れる形ならまだしも，好きな形を作るのには少々無理があった。メタボールは，近づけると変形・融合して思いもよらない曲面に変形するので，生き物の身体といった複雑な曲面に大きな力を発揮する。

メタボールは，そのサイズとほかのメタボールとの近づけ方で形が決まる。それゆえ，形を思いどおりに制御するのは難しい。メタボールの原理について，多少なりとも理解していることが必要だろう。たとえばサンプルのタイヤなど，2個のメタボールで作っているのだが，どうくっつけたらドーナツ形になるのだろう（論理演算は使っていない）。これを理解するのに，多少の知識がいるということなのだ。というわけだから，正式にレビューするときになったら，難解になるのを覚悟で，じっくり解説したいと思う次第である。メタボールは，レイトレーシングで扱えるプリミティブのうちで，いちばん扱いが難しく，そしていちばん面白いものだと思うからだ。

メタボールはふつうのプリミティブより数段重い処理になるのだが，速度はマル。試用バージョンはトランスピュータボード用だが，50個近くメタボールを使っているこのサンプルの場合，512×512ドットで5時間だからかなりのレベルだ。

なお，このボードとソフト込みで定価61万円だが，いまなら特別定価45万円とのこと。今回は未完成だったメタボール版も間もなくリリースされる予定である。(A.T.)

これでプリミティブは50個程度

特集1

X1プログラミングガイドブック

最近はX１用に市販されるアプリケーションも少なくなった。メーカーのサポートも期待できない。といっても，特に驚くにはあたらない。すでに多くのマシンがたどってきた道筋である。むしろ，ほかのマシンに比べて，ずいぶん健闘したといえるだろう。ではもう，X１は終わりなのか？　否，パソコンというのはそんなものじゃないし，そうあってはいけない。ここでユーザーの真価が問われることになるのだが，X１関係の投稿状況を見ていると少々情けないものがある。

原点に戻って，X１発売の頃を思い起こそう。重いBASIC，まったくわからないI/O，どこにもないアセンブラ。G-RAM構成さえ手探りで探っていかなければならなかった。まったく何度画面を真っ白にしたことか。しかし使い込むほど，実は非常に素直なマシンであることがわかってくる。苦労の分だけ愛着がわき，ユーザーに愛されるマシンとして，X１はひとつの神話を作った。初期のパワフルなユーザーがそのままX68000に移行したおかげで，現在のX１勢力はやや弱いものになっている。残されたユーザーはX68000を手に入れるまで眠りにつくのか？　もはや「黄金の時代」は過ぎ去った。ユーザーが自分で動かなければ，マシンは応えてくれない。しかし，こういった，いわば「青銅の時代」こそが，自然なパーソナルコンピューティングを教えてくれるのではないだろうか。

CONTENTS

特集1

序論 X1の正しい使い方

Urakawa Hiroyuki
浦川 博之

「X1は死んだのか？」などと間抜けなことをいっちゃいけない。ユーザーが生きている限りX1は不滅だ。しかし，パソコンのなかでの立場が変わってきたことは確かだ。16ビット全盛の現在，いかにX1を使うべきか。X1ユーザーによるメッセージを送ろう。

時代はいまや……

「時代はいまや16ビットだ」

あ，痛て。痛てて。ものを投げるな，ものを。だって，8ビットパソコンのリーダーだったPC-8801はPC-9801に吸収されちゃったし，そのPC-9801は累計出荷が214万台。最近では東芝から10万円台の16ビットラップトップまで登場したじゃないか。ワープロやゲームマシン，パソコンに比べれば低級に見えるようなものまで68000が採用されている（しかも決して高級機としてではない）ことからしても，いまや16ビットCPUが大衆のものになっているといわざるをえないだろう。

それどころかSONYから発売されたAXは32ビット（80386SX）で30万円のカベを破ってしまった。いわんや，もう上級機は32ビットで当然というところまできている。さ来年あたりは，

「時代はいまや32ビットだ」

といってもおかしくない時代になってるかもしれない。

と，しゃあしゃあといってのけるけど，X1の登場の頃は16ビットCPUと聞いただけで「うわぁ，スッゴイなぁ」と漠然とした尊敬と畏怖の念を抱いていたのが，いまや「32ビット，ふ～ん」という調子なんだから，この業界ほど昔の価値観があてにならないところも珍しい。

16ビット時代のコンピューティング

ところが。16ビット時代といわれて久しいクセして，その機能を使いこなす人々がプロに限られ，個人でプログラミングをする姿があまり見られない。

少なくとも8ビット全盛の頃はいまよりプログラミングそれ自体を楽しむ人々が多かったし，実際，いまも8ビットに残っているというのに。8ビットパソコンのすべてが16ビットパソコンに継承されているわけではないということだろうか。

このあたりはコンピュータの性格を2つに分けてみればわかりやすい。ひとつはPC-9801に代表される，ソフトウェアプレイヤーとしての性格。もうひとつは自分でプログラミングすること自体を目的とするプログラミングマシンとしての面だ。

つまり，こういうことだ。確かにソフトハウスの送り出すアプリケーション，ビジネスはもちろん，最近ではゲームまで既存の8ビットCPUでこなせる範囲を超えてしまっている。だが，それとともに急激にメモリ容量，スピードなどの処理能力を上げたハードウェアが，単にアプリケーションの受け皿としての面しか成長していないということだ。

ハードウェアの成長は，よりアプリケーションを効率よく動かすため，「コンピュータのことなんか知らなくてもいいや」という人たちのためであって，東海の小島の磯の白砂でコンピュータと戯れていたい人のことはあまり考えていない。思わず泣き濡れてしまいたくなる話だ。

いかに扱いがややこしくなろうとも，まずスピード，次にスピード，最後にメモリ。じゃ，あとのことはソフトハウスさんよろしくね，とまぁそんな態度なわけだ。

EMSボードなんかがそのいい例で，なにが悲しゅうて80386を搭載していながら16Kバイト単位切り換えのRAMディスク（でもないか）を付けねばならんのと思うと涙が出てくる。ビジネスツールとして見るなら商品性を高める当然の経営戦略だろうが，ハードが友だちという人から見ればあまりにもタコだ。

さすがにX68000は16ビットパソコンの中でも例外的だ。そういったホビーユーザーのことも考えているようで，MPU68000の採用に始まって，メモリ配置からX-BASICの開発まで，マシンのコンセプトに「プログラミングして楽しいマシン」を据え，その障害になるようなことは極力避けているような姿勢が感じられるし，実際その姿勢はユーザーの心をつかんでいる。パソコンをソフトウェアプレイヤーとしてしかとらえられない人には，市販ソフトが1本もないマシンに飛びつく人がさぞや奇異なものに映ったに違いない。

まぁ，X68000は魅力的なマシンだし，現状のMZ，X1ユーザーはどんどんX68000に移行している。これは無理からぬことではあろう。しかし，「プログラミングして“真に”楽しい」のは，すでにX68000に出会う前からその楽しさを知っている人々に限られるのではないだろうか。

確かにX68000からでも入門はできるし，マシン語にしろDOSにしろハードまわりにしろ8ビットパソコンより，ずうっと口当たりはよいだろう。しかし「使いこなしたぞ！」といえるゴールは8ビットに比べて遙かに遠い。君はひとりでスプライト128個を「使いこなした」ソフトが作れるだろうか？ 16ビット機の性能があらゆる面で8ビット機を凌駕していても，8ビットパソコンにあったあの“使いこなす楽しみ”のほうは，必ずしもきれいに受け継がれているわけではないのだ。

8ビット機の地位は？

というわけで，さあX1ユーザーの皆さんお待たせしました，やっぱりプログラミングの楽しみを味わうなら，8ビットは必修科目だねという話だ。

なんといっても8ビットは，

メモリが少ない
グラフィック能力が低い
Z80などCPUが低級

という，数々の利点がある。

待て。待て。ものを投げるんじゃない。逆にいえばこれらはみな，マシンの天井に手が届きやすく，そこから工夫して自分のやりたいことを実現するための技を磨けるということでもあるのだ。いい換えれば，

手頃なRAM容量
軽快な画面表示

シンプルなアドレッシングモード
となる（昔は「高級機に負けない価格」というオチもあった）。

マシンの真の性能は，限界と思われる壁をひとつずつ突破していくことによってのみ発揮される。「限界は超えるためにある」とはいえ，普通の人間にそうそう超えられるわけないじゃないか，という意見もあるかもしれない。でも別に超絶的技巧や超人的努力を要求しているわけじゃない。それぞれの人にはそれぞれの限界がある。それを超えるとまた次の限界が見えてくる。これを超えようとし続けることが，すなわち限界を超えることなんじゃないかと思う。いつだって，マシンより人間側の度量がマシンの力を決めている。

64Kバイトに640×200ドットの画面でもよほどのエキスパートになるか，大作を組むかしないと窮屈にはならない。もし窮屈と感じたら，まず自分のプログラムの効率の悪い部分を探してなんとかすることを考えたほうが正解だ。

8ビットパソコンは初めてそういうコンピュータとの付き合いを覚えるところ，そして「なんとかする」そのなんとかしかたを覚えるところとしての地位はまだ失っていない。

プログラミングの楽しみ

X1の場合，登場したての頃からずーっとその性格を保持し続けて，ユーザーに直接いじられることを考えたアーキテクチャを貫いている。登場期に，PCGやPSGを装備していることから「ゲームマシンだ」という批判を受けたこともあったが，これは見た目が派手だからということで，それらの機能はいまでもユーザーフレンドリなものとして使われているのだから，完全なソフトウェアプレイヤーになっているというわけではない。ソフトウェアのレベルに合わせようと頑張って，ごちゃごちゃの構造になってしまったものよりは，ずっとホビーユーザー向けとしての地位は固い。

Oh!Xという雑誌が成立している（おかげさまで）だけでも，X1ユーザーにプログラミングを楽しむ人々が多いことがわかるだろう。X1を持っていながら自作プログラムの作成も移植も改造もやったことがないというのは，ちょっと反則だぞ。

そういう楽しみのわかる人には，それ特有の症状が発生する。そのひとつが「自分のプログラムに対する他人にはわからない愛着」である。

君のまわりにもいるだろう，ずーっと昔のゲームをいまだに改良し続けている人。もう市販のものに比べればとんでもなく機能が低いユーティリティを残していて，人になにかいわれると，「いや，これはいつか改良を加えて，○○並にしてやるんだ」といっている人。

はたから見れば，いつまでも固執していないで，作り直したほうが早いのに……と，思いたくもなるが，その人にとっては，ソフトウェアによって達成される目的もさることながら，そこに到達するまでの過程が大切なのだ。だから必ずしも100％のものが完成しなくても構わないし，できたらできたでまた嬉しい。

市販のアプリケーションと自作のものを比べたら雲泥の差があるだろうが，自作した本人の中では，市販ソフトは市販ソフト，自作ソフトは自作ソフトというまったく別の尺度が成立してしまっているのだ。

だから他人のソフトウェアでアニメーションをやっていても「ふーん，そだね」程度の反応しか示さなかった人が，自分で作ったとなると「ねぇねぇねぇ見て見て見て。アニメーションするんだアニメーション」と，態度が一変する。えてして見せた人から「ふーん，そだね」の逆襲をあびせられて，「いやー，たいしたもんでもないんだけどさ，ハハハ」とかなんとかいいながら心中深く傷つくというのが相場になってはいるが。まぁでも，市販ソフトと，自作ソフトの間に別々の尺度が確立したということは，プログラミングの楽しさがわかったということでもあるのだ。

コンピュータって？

自作のソフトに，そのレベルに関わらず愛着がわくのはなぜだろう。自作ソフトは市販ソフトと違って動くのが当たり前というものではない。もしかしたら完成しなかったかもしれない，という，数々の「未完成への確率」をくぐり抜けてここまできたんだということ，さらにそこまでたどり着かせたのは自分なんだという自覚があるからだ。そういう経験を積んで，しだいにコンピュータを操る術を身につけていく。

いったんそれがわかれば，X1だろうとX68000だろうと，PC-9801だって，ソフトウェアプレイヤーとして使うときはあるにしても，機械を主体的に使いこなす態度を忘れることはないだろう。逆にいえば，「X1も使えねーのに，X68000さえあればなんとかなるわきゃねーだろっ」ということになる。どーだ，怖い話だろう。

コンピュータを自力で従わせた経験のない人は，いつまでたってもアプリケーションユーザーからは抜けられない。いったんそういう経験を持てば，そのあとは機種を問わず，すぐに使いこなせるようになるから（ただし好みは出る），本当にコンピュータの“ユーザー”になろうというのなら，X68000よりは，むしろ限界の低いX1のほうが向いているということになる（勘違いされると困るのだが，限界が低いというのは見通しがいいということであって，面白いことができないというのとはまったく別次元の話だ）。X1をクリアできて初めてX68000の冥利がわかるのだ。

市販のソフトを使うことだけを考えて購入した人には，とやかくいえた筋合いじゃないけど，パソコンには確実にプログラミングの世界があるのだから，そのことを知っていて購入した人は少しでも関心を持ってほしい。特にルールはないとはいえ，やっぱり運転手つきでポルシェに乗るのは恥ずかしいと思うんじゃないかな。パソコンも結局はこれと同じようなものだ。

要は態度，チャレンジ精神の問題なのだ。安易に機能のせいにしてX68000に移ったって絶対にもっと手ひどくフラれる。結局，コンピュータというものは「なにかをしてくれるもの」じゃなくて「なにかをするためのもの」なのだから，なにもしようとしなければ，当然なにもできない。

8ビットだからX1は使うのがやさしいだろうと思ったら，これもまたフラれることになる。確かにPCGやRAM配分がユーザーフレンドリといえるけど，自分がいいかげんではつっけんどんにしか応対してくれないのは当たり前だ。その代わり，自分がしたいことをきちんとまとめて伝えれば，それに見合った満足を返してくれるのがいいところ。

まず，なにかをしたいと思うこと。すると，たいてい壁にぶつかる。でも，それは「X1の壁」ではない。そう信じて壁を突き破っていく。これを繰り返していくことでやっとX1の性能が発揮されるのだ。

まだまだX1にはあなたの味わったことのない感動や性能があるのかもしれない。この特集を読んで，「えっ，そうなの？」と思う部分があったら，もう一度X1の前に座ってみることをお勧めする。

そして，プログラミングを終えたうえで，「フッ，オレに使われるとは幸運な機械だぜ」という，はた目には不気味な笑みをもらしてもらいたいのだ。

特集1

ハードウェアから見たX1

私がX1にこだわるわけ

グラフィックもミュージックもマニアタイプでこなしてしまう三沢氏が，何年たっても色あせないマシンの魅力を語ってくれました。その充実したハードに触れたら，あなたはもうX1をやめられない。

Misawa Kazuhiko
三沢 和彦

X1マニアタイプがデビューしてから6年半がたちました。私自身もマニアタイプのユーザーであり，私のパソコン歴はX1の歴史と共にあったといえます。いまではX68000が登場し，X1の影が薄くなってしまったようですが，正直なところ，パソコンを隅々まで使いこなすことを目標にする人には，X1はお勧めのマシンだと思います。ここでは，私がそう考える理由をX1のハードの面からお話ししましょう。

進化するパソコン

6年以上も前のパソコンをいまさら……と即断する人に限って，その人のマシンは高級ファミコンかワープロ専用機にしかなっていないことがあります。私自身，X68000とX1マニアタイプ両方のユーザーですが，X1だけでもあまり困ることはありません。

X1のハードウェアは，基本性能では一般の16ビット機と比べても，劣るものではないでしょう。X1turboならば，

・640×400ドット8色グラフィック
・漢字VRAMによる漢字表示
・32KバイトBIOS ROMによるシステム
・バンク切り換えにより最大576Kバイトまでのメモリをサポート

というような機能も簡単に実現してしまうのです。また，FM音源ボード，カラーイメージボード，立体映像ボード，テロッパなどの豊富な周辺機器もサポートしていますから，AV関係でも高度な処理が可能です。MIDIボードの製作記事もOh!Xに掲載され，MIDIライフもスタートしました。

もちろん標準装備のHuBASICは，X1のハードウェアの性能をフルに発揮させてくれる，使い勝手も悪くないBASICです。また，CP/Mとそのランゲージシリーズが極めて安価なので，FORTRAN，Cなどの高級言語も安心して使えます。

マシン語の開発環境では，CP/M上のMACRO80もですが，S-OS上のZEDAを始めとするアプリケーションのラインナップが強力です。私もマシン語開発はすべてS-OS上で行っています。マルチウィンドウエディタWINERの威力には，16ビット機ユーザーの友人も感心していました。HuBASICでメインプログラムを組み，マシン語サブルーチンをS-OS上で組む。私はこのやり方で十分満足しています。

もちろん，ハードウェアの基本性能，市販ソフトウェアの質，総合的なプログラム開発環境，どれをとっても後発のX68000のほうが優れているに決まっています。しかし，私にとってはX1を6年間使いこなしたあとにX68000を手に入れたからこそ意味があるのです。

それは，X1がユーザーを成長させ，X1自身も進化し続けるパソコンだったからでしょう。

クリーンコンピュータ

では，X1はどんな面でユーザーを成長させてくれたのでしょうか。それは次のようにいくつかのポイントとして挙げることができます。

1) クリーンコンピュータ
2) I/O空間に配置されたリニアなG-RAM
3) サブCPU，PCGやPSGなどの周辺ハードの充実

2)，3)の項目はハードウェアの具体的な特長であり，これらについてはあとで説明します。重要なのは1)のクリーンコンピュータ思想であり，これは根本的なコンピュータに対する考え方がユーザーフレンドリなものだったということです。現在なおX1が現役でいられる理由は，すべてこの思想に帰せるのではないでしょうか。

クリーンコンピュータというのは，ハード自体がROMの形ではシステムソフトを持たず，コンピュータの起動時に外部記憶装置から読み込んでくる，というものです。現在はMS-DOSマシンもX68000もこの方式であり，もはやクリーンコンピュータ以外のコンピュータのほうが一般的ではないかもしれません。しかし，X1シリーズの初期（もちろん，最初のMZ-80Kからすでにそうだった）には，カセットテープから4分半もかけてBASICをロードするのは狂気の沙汰ほどに思われていました。しかしながら，このクリーン設計のおかげでBASICがバージョンアップしてもマシンを買い換えたりしないですんだのです。

クリーン設計のよさはそれだけではありません。システムが外部から供給されるということは，その中身を書き換えられるということです。本来，システムを書き換えるのは，暴走の危険もあり掟破りなのですが，あえて挑戦することで学ぶことも多いわけです。

たとえば，HuBASICのメモリマップを見てください（図1）。これによるとHuBASICがIOCSとインタプリタのふたつの部分に分かれていることがわかります。IOCSとはInput Output Control Systemの略で，ハードウェアの入出力関係（ディスプレイへの表示やキーボードからの入力）の基本的なサブルーチンを集めたものです。

BASICインタプリタは，命令を解釈すると，このIOCSルーチンへの呼び出しを組み合わせて実行します。X1turboでは，この部分だけROM化してあります。MS-DOSやCP/MといったOS (Operating System)も，このIOCSと基本的には同じものなのです。

このIOCSのサブルーチンは，マシン語プログラムからも呼び出して実行させることができます。ハードウェアに密着した部分はすべてIOCS内で処理してくれるので，マシン語プログラムといっても簡単なものですむわけです。こう考えると，IOCSだけ切り離して使ってもよいわけです。

S-OSはHuBASICからこのIOCSを切り離し，IOCSルーチンを2，3まとめて処理するサブルーチンとして整備した部分を，つけ足したものです。ここで，IOCSだけ切り離すのは，システムが書き換えられるク

リーン設計だったからできたのだ，という点が重要です。

FM音源やMIDI対応のMML（MusicBASICなど）もHuBASICを書き換えています。これは，インタプリタ部分に拡張MMLの処理ルーチンを書き加えています。

さて，皆さんにもシステムの書き換えを実体験してもらうために，サンプルをいくつか用意しました。まずは，ファンクションキーの登録内容の書き換えです。ファンクションキーはDEF KEY命令で変更できますが，BASICを立ち上げるたびに内容が元に戻ってしまいます。それなら，ディスクの中身を書き換えてしまおう，というわけです。BASICからDEF KEY命令で変更したあと，リスト1を実行させてみてください。次からは起動時にすでにお望みの内容に変わっています。ただし，リスト1中のDEVO$命令は，無防備にディスクの内容を書き換えてしまうので，万一に備え，必ずバックアップを取ってから実行してください。

次に，よく見かけるものですが，テンキーを16進キーに変更するプログラムをリスト2に挙げます。BASICレベルでキーの変更ができ，MON命令でモニタに入れば，そのままマシン語入力が16進キーで行えます。ただし，この16進キー変更は，BASICを立ち上げ直すたびに実行させるようにします。というのは，テンキーの+，－，＊，/，＝などの記号も一般的によく使われるからです。

これらのシステム書き換えは，BASICがROMの中にあるときには不可能なワザです。これが可能なことで，ちょっとした工夫をする知恵がつくようになる，これはユーザーが成長するということの意味するもののひとつであり，クリーン思想はユーザーの成長の一助となっているのがおわかりいただけるでしょう。

X1のハードの特長

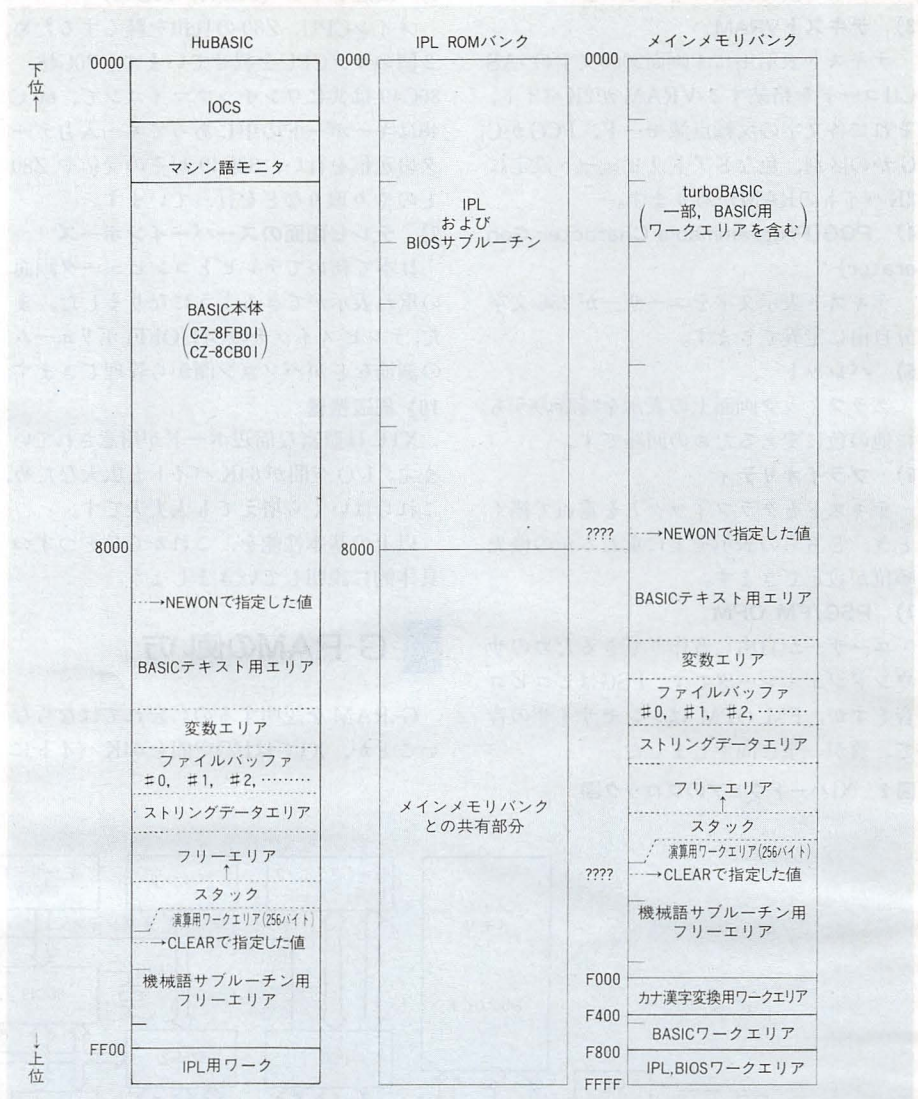
次にX1のハードウェアのブロック図を図2に示し，ハードの特長をひととおり説明します。

1） メインメモリ

前述したように，64KバイトオールRAMのクリーン設計です。

2） G-RAM

X1の特長のひとつである48KバイトのG-RAMは，I/O空間にリニアに配置されています。X1turboでは解像度が倍になるので96Kバイト必要ですが，これは48Kずつの

図1 HuBASICのメモリマップ

リスト1 ファンクションキーの書き換え

```
10 ' SAVE "1:SAMPLE1.BAS
20 KEYLIST : PRINT
30 INPUT "DISK WRITE OK ? [Y/N]",Z$
40 IF Z$="Y" OR Z$="y" THEN 50 ELSE END
50 ADD=&HF20
60 ADD=INT(ADD/256)*256
70 REC=ADD/256+&H20
80 DEVI$ "0:",REC,A$,B$
90 IF LEFT$(A$,16)<>HEXCHR$("00000100000000000000001000000000")
THEN PRINT "ERROR" : BEEP : END
100 C$=MEM$(ADD,128) : D$=MEM$(ADD+128,128)
110 DEVO$ "0:",&H2F,C$,D$
120 PRINT "DONE "
```

リスト2 テンキーを16進キーに

```
10 '  SAVE "1:SAMPLE2.BAS"
20 CLEAR &HFD00
30 INPUT "HEX KEY OR NORMAL KEY ? [H/N] ";Z$
40 IF Z$="H" OR Z$="h" THEN 50 ELSE IF Z$="N" OR Z$="n" THEN 80
ELSE END
50 POKE &H3B4,&HC3,&H0,&HFD
60 MEM$(&HFD00,48)=HEXCHR$("7AE602CAB7037B1E41FE2E28201CFE3D281B
1CFE2B28161CFE2D28111CFE2A280C1CFE2F2807FE2C20023E205FC3B903")
70 END
80 POKE &H3B4,&H7A,&HE6,&H2
```

バンク切り換え方式になっています。

3) テキストVRAM

テキスト表示用に1画面2000文字の ASCIIコードを格納するVRAMが2Kバイト，それに各文字の反転点滅モード，PCGかCGかの区別，色などアトリビュート設定に2KバイトのRAMがあります。

4) PCG(Programmable Character Generator)

テキスト表示文字をユーザーが256文字分自由に定義できます。

5) パレット

グラフィック画面上の表示を瞬時のうちに他の色に変えるための回路です。

6) プライオリティ

テキストとグラフィックとを重ねて描くとき，どちらの表示を上に重ねるかの優先順位が設定できます。

7) PSG/FM OPM

ユーザーが自由に音作りできるためのサウンドジェネレータです。PSGはピコピコ音ですが，FM OPMはシンセサイザの音で，質が一挙に向上しました。

8) 周辺チップ（80C49，80C48）

メインCPU，Z80の負担を軽くするため，2個のサブCPUを載せています。80C48，80C49は共にワンチップマイコンで，80C48はキーボードの中にあってキー入力データの送信を行い，80C49がその受信やZ80とのやり取りなどを行っています。

9) テレビ画面のスーパーインポーズ

日本で初めてテレビとコンピュータ画面の重ね表示ができるようになりました。また，テレビスイッチのON/OFF，ボリュームの調節などがパソコン側から管理できます。

10) 周辺機器

X1には豊富な周辺ボードが用意されています。I/O空間が64Kバイトと広大なため，これらはいくら増えても大丈夫です。

以上の基本性能を，これからひとつずつ具体的に説明していきましょう。

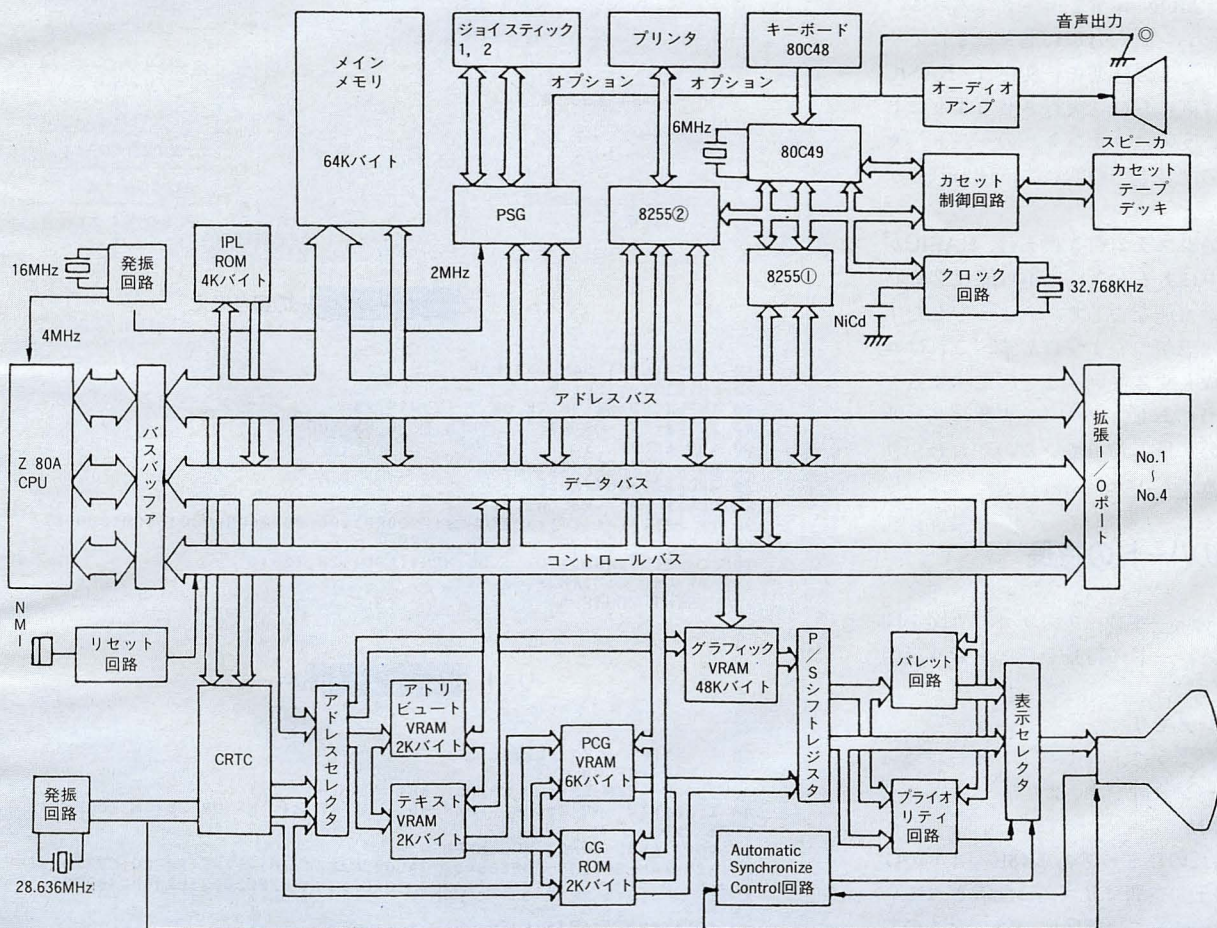
G-RAMの使い方

G-RAMを説明するのに忘れてはならないことが，X1ではI/O空間を64Kバイトに拡張しているということです。X1以前の機種では，48KバイトのG-RAMを配置するのにメインメモリの一部をバンク切り換えすることによってアクセスしていましたが，それではプログラムがバンクの裏に隠れてしまうときには正常にアクセスできないことがありました。

それに対し，G-RAM専用メモリをメインメモリと切り離して考えられるX1方式では，わかりやすさが格段に違います。このようにG-RAMを外部メモリとして配置できたのもI/O空間が64Kバイトになっていたからです。

X1では，グラフィック画面1ドットがG-RAM上の1ビット（ただし，RGB合成のカラーを考えればそれぞれ1ビットずつ計3ビット）に1対1で対応しています。画面の解像度によって複数ページが使用できるようにするために，画面上の座標値(X,Y)とG-RAMのアドレスの間には多少複雑な関係式があります。

図3にI/Oマップを示しておきますが，これによるとG-RAMのアドレスはBlueが

図2 X1ハードウェアのブロック図

4000H～7FFFH，Redが8000H～BFFFH，GreenはC000H～FFFFHになっています。横8ドットひとまとまりで1バイトデータになっていて(図4)，それぞれのバイトのアドレスは，640×200の解像度では座標（X，Y）に対して，

(OFFSET)+(X¥8)+((Y MOD 8)×(800H))+(Y¥8)×80D

で算出されます。OFFSETの値は，Blueなら4000H，Redなら8000H，GreenならC000Hです。320×200の場合では×80Dを×40Dに変えればOKです。

さらにX1turboの場合は640×400の高解像度もありますが，このときはYの値が偶数ならバンク0上，奇数ならバンク1上，というように1列ずつバンクが入れ替わっています。具体的には（Y MOD 1）をバンクNo.にして(Y¥2)の値を上の式のYに代入すればいいのです。すなわち，X1用のグラフィックデータを，X1turboでは1列おきに表示することによってコンパチビリティを保っているわけです。

しかし，このあたりの事情は，HuBASICのグラフィック命令を使っているかぎりあまり重要ではありません。大切なのは，このG-RAMがI/O空間上にある外部RAMだというコンセプトです。

現在では16ビット機などでRAMディスクというものが普及していますが，このG-RAMはまさにRAMディスクそのものといえます。これはHuBASICでもOPTION SCREEN命令でサポートしています。RAMディスクは普通のフロッピーディスクと同じようにアクセスでき，しかもスピードが格段に速いとあって便利なデバイスです。

X1ではすでに6年前から，このデバイスを取り入れていたのです。G-RAMという名でグラフィック専用だと思われるものを，データ格納用のメモリに転用する，という工夫が可能だったわけです。

リスト3は，X1の48KバイトのG-RAMデータをディスクにまるごとセーブ，あるいはディスクからロードするプログラムです。このプログラムのミソは，グラフィック用に使用していたG-RAMをOPTION SCREEN 2でRAMディスクだと考え直し，データをべったり転送する点です。Oh!Xのバックナンバーでもどこかで出てきたアイデアでしょう。オールBASICなのであまりスマートではありませんが，初心者の人へのひとつのサンプルとして載せておきます。

グラフィック特殊制御とPCG

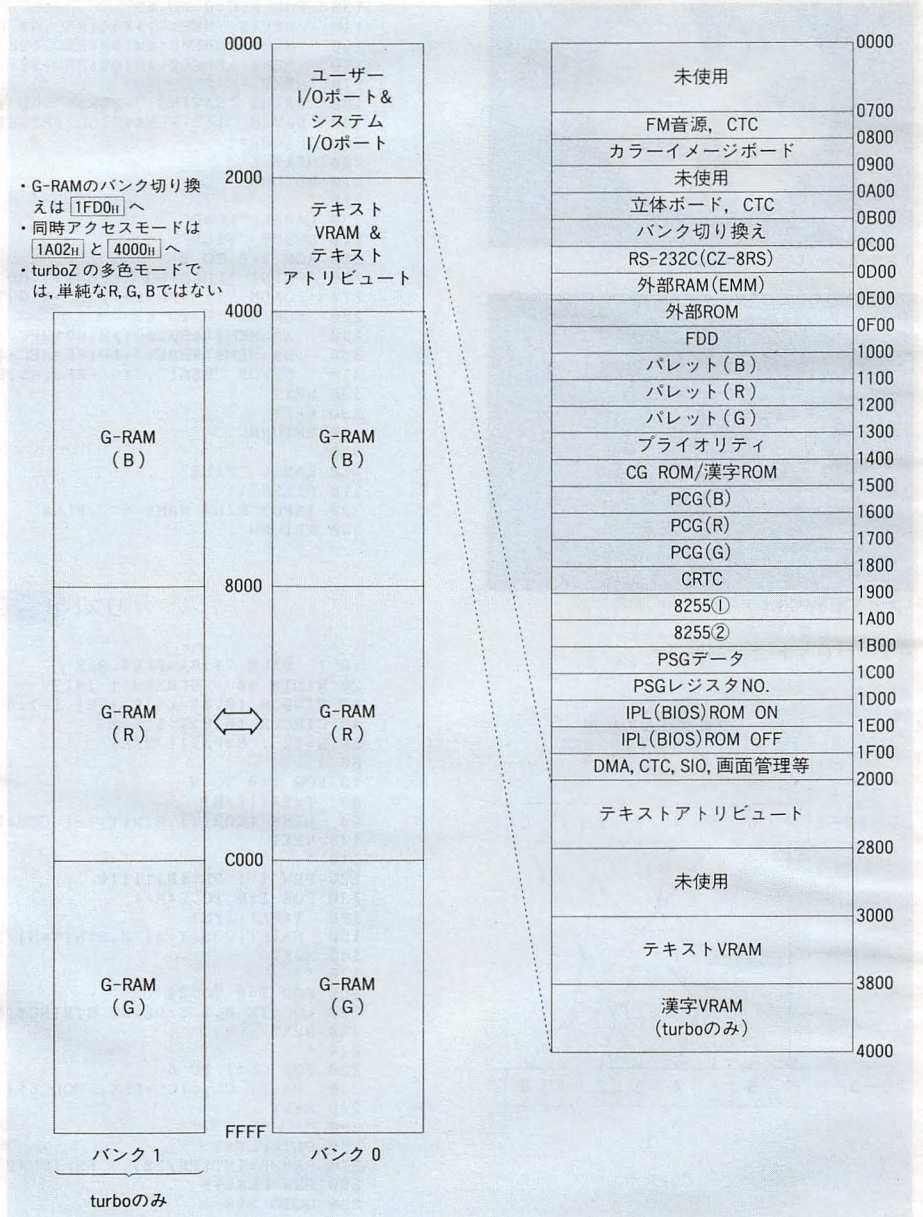
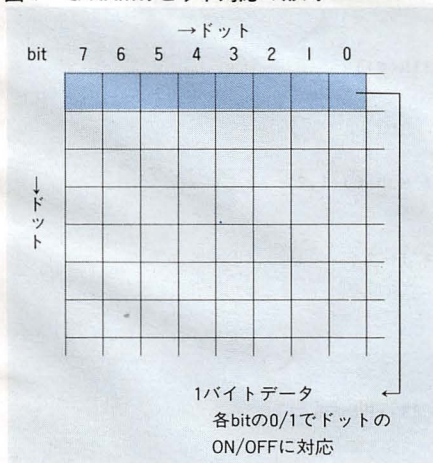
X1のグラフィック機能が優れている点として，パレットおよびプライオリティがあります。

パレット機能とは，グラフィックVRAM上のデータを変更せずに瞬時に色を変えるものです。X1では，カラーコードは直接表示色にはなっておらず，カラーコードと表示色とを対応づける回路を通して表示しています。図5を見てください。この対応を変えることによって，同じカラーコード(パレットコード）でも色を変えることができるのです。

プライオリティ機能とは，テキスト画面とグラフィック画面とを両方重ねて表示するときに，グラフィック画面上の各パレットコードごとにテキストとグラフィックのどちらを前に表示するかが指定できるものです。

ここで重要なのは，プライオリティが色

図4　G-RAMのビット対応の形式

図3　I/Oマップ

単位で指定できるという点です。これによって画面に奥行を持たせることが可能になります。

グラフィックは「百聞は一見にしかず」ですから，とりあえずリスト4を実行させてみてください。円板が時計回りに回転しているように見えるでしょう。これが，パレット機能で瞬時に色を入れ替えて見せているのです。同じプログラムをPAINT命令でひとつずつ塗り変えていくようにすると，これほどなめらかな動きにはなってくれません。さらによく見ると，＊印の並んだ模様が反時計回りに回転しているように見えませんか。しかもこの模様は，円板の外側(黒い部分)からはみ出していません。＊印はテキスト画面の文字を並べて表示しているのですが，円板の内側だけ表示させるのがプライオリティ機能なのです。

同じことを，座標位置を計算して円板の内側ならON，外側ならOFF，というようにいちいち判断させるのはプログラムが複雑になるだけです。さらに，扇形内だけの模様を回転させるのは，プライオリティの切り換えを使わなければとても困難です。

これらの特殊表示機能はPCGと組み合わせることによって威力倍増します。

そもそもテキスト画面は，256文字の中からそのASCIIコードを指定して文字表示を行います。テキストVRAMは80×25文字の画面1文字ごとにその表示内容をASCIIコード1バイトで記憶しておき，実際にCRTに表示するときはそれぞれのASCIIコー

リスト4のパレット機能の実行

リスト5 簡易PCGエディタ

図5　パレット

リスト3　G-RAMデータの転送

```
10 '  SAVE "1:SAMPLE3.BAS
20 CLEAR &HBE00
30 CLS : OPTION SCREEN 2
40 '
50 INPUT "GRAM DATA SAVE OR LOAD ? [SAVE=1/LOAD=2] ",Z$
60 IF Z$="1" THEN GOSUB "SAVE" ELSE IF Z$="2" THEN GOSUB "LOAD"
ELSE END
70 OPTION SCREEN 1 : INIT
80 CLS : END
90 '
100 LABEL "SAVE"
110 GOSUB "FILE"
120 FOR I=0 TO 2
130  FOR REC=0 TO 63
140   DEVI$ "MEM:",I*64+REC,A$,B$
150   MEM$(&HBE00+&H100*REC,128)=A$
160   MEM$(&HBE00+&H100*REC+128,128)=B$
170  NEXT
180  PRINT "SAVING "+FLN$+".GD"+RIGHT$(STR$(I+1),1)
190  SAVEM "1:"+FLN$+".GD"+RIGHT$(STR$(I+1),1),&HBE00,&HFDFF,&H4
000+&H4000*I
200 NEXT
210 RETURN
220 '
230 LABEL "LOAD"
240 GOSUB "FILE"
250 FOR I=0 TO 2
260  PRINT "LOADING "+FLN$+".GD"+RIGHT$(STR$(I+1),1)
270  LOADM "1:"+FLN$+".GD"+RIGHT$(STR$(I+1),1),&HBE00
280  FOR REC=0 TO 63
290   A$=MEM$(&HBE00+&H100*REC,128)
300   B$=MEM$(&HBE00+&H100*REC+128,128)
310   DEVO$ "MEM:",I*64+REC,A$,B$
320  NEXT
330 NEXT
340 RETURN
350 '
360 LABEL "FILE"
370 FILES"1:
380 INPUT"FILE NAME = ",FLN$
390 RETURN
```

リスト4　パレット機能の例

```
10 '  SAVE "1:SAMPLE4.BAS
20 WIDTH 80 : SCREEN : INIT
30 WINDOW (0,0)-(639,199),(-1.5,1)-(1.5,-1)
40 CIRCLE (0,0),.5,7,2
50 N=12 : S=PAI(1/2/N)
60 '
70 FOR I=0 TO N
80  T=PAI(I/N)
90  LINE (COS(T),SIN(T))-(-COS(T),-SIN(T))
100 NEXT
110 '
120 PRW 1 : PR=&B111110
130 FOR I=0 TO 2*N-1
140  T=PAI(I/N)
150  PAINT(COS(T+S)/2,SIN(T+S)/2),(I MOD 6)+1,7
160 NEXT
170 '
180 FOR I=0 TO 24
190 LOCATE 0,I : PRINT STRING$(80,"*");
200 NEXT
210 '
220 FOR CL=1 TO 6
230  PALET CL,((CL+PLT) MOD 6)+1
240 NEXT
250 '
260 PLT=PLT+1
270 PRMSB=INT(PR/32) : PR=(PR*2 MOD 64)+PRMSB
280 PRW PR*2+1
290 GOTO 220
```

ドに対応するパターンをROMから読み出してくるのです。

通常は，そのROM(Character Generator)には0～9の数字，A～Zとa～zのアルファベット，+，-，/，*，=などの記号のパターンが書き込まれています。X1は，PCG RAMを持ち，その256文字分のパターンをそっくりユーザー定義文字に入れ替えることができます。そしてその文字パターンは，8×8ドットずつ，個々に色を指定できるのです。

すなわち，8×8ドットのグラフィックパターンをひとまとまりとして定義しておき，64ドットのデータを書き込むかわりに1バイトのコードですませられるというわけです。ですから，それぞれのパターンにインベーダーや戦闘機やロケット弾などのゲームキャラクタを定義しておけば，高速表示が可能なのでゲームが作りやすいのです。しかも，PCGをプライオリティと組み合わせれば，グラフィック画面で背景を描いておき，奥行のあるスクロールを簡単に実行させることもできます。PCGの使い方は，亀田氏の記事にもなっていますので参考にしてください。ここではリスト5に簡単なPCGエディタをサンプルとして挙げておきます。

サウンドジェネレータ

X1には標準でPSGが載っていますが，FM音源が普及した現在では，もはや時代遅れかもしれません。FMボードはぜひ載せておきたいものです。

このFM音源は，オペレータと呼ばれるサイン波発生器を4つ（高級なものでは6つ，あるいは最近ではサイン波だけでなく他の波形も発生できるものが主流）組み合わせ，それらの基本波形を足したり掛けたりして任意の波形を作り出すものです。FM音源は，ヤマハのシンセサイザに搭載されており，プロのミュージシャンにも広く使われています。

X1用FM音源ボード付属のソフトVIPには，音色データが200種類プリセットされていて，それらは自由にエディットできます。また，FM音源による演奏もOh!X掲載の，MusicBASICなどがMMLをサポートしているので，BASICプログラムからも処理できる環境になっています。ここではサンプルプログラムは挙げませんが，ぜひFM音源用MMLを試してみてください。

ところで，サウンドジェネレータとは直接関係ないのですが，PSGとFM音源ボードとに関連して，ユーザーに便利な機能を紹介しておきましょう。PSGにあるジョイスティックポートと，FM音源ボードにあるCTC(Counter Timer Circuit)です。

ジョイスティックポートは，PSGについている汎用入出力ポートのことです。このジョイスティック端子は9ピンのコネクタになっていますが(図6)，このうち1～7，9ピンはI/Oポートにつながっていて，CPUと入出力のやり取りが可能です。

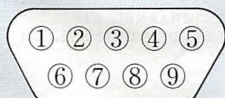
すなわちCPUからデータを送り出すとこの端子にH/Lのデータが出力され，逆にこの端子をそれぞれH/Lに設定しておくと，CPUからその状態が読み取れるわけです。ジョイスティックはこのI/O入力の機能を使っていることになります。

自分で簡単なハードを製作してこのポートに接続すれば，高価なI/Oボードを使わなくても周辺機器としてつなげることができます。たとえば，Oh!X 6月号の「学習リモコンの製作」では，X68000を使ってですが，ジョイスティック端子とリモコンをつないでいます。また，リモコンロボットをこのポートにつないで，パソコンからロボットを制御する試みもあります。このポー

図6　ジョイスティック端子

（本体後面側から見た図）

リスト5　簡易PCGエディタ

```
10 '  SAVE "1:SAMPLE5.BAS
20 WIDTH40:COLOR7,0:SCREEN 0,0,0
30 CGEN0:WINDOW:CLEAR : INIT : CLS 4
40 DIMA%(23,23),B%(2):DEFINTA-Z
50 FOR I=8 TO 16:C=1
60 IF(I MOD 8)=0 THEN C=5
70 LINE(I*8,8*8)-(I*8,16*8),PSET,C
80 LINE(8*8,I*8)-(16*8,I*8),PSET,C
90 NEXT
100 LOCATE24,0:PRINT"*PCG EDITOR *"
110 LOCATE27,6:PRINT#0 CHR$(&H1E)
120 LOCATE24,8:PRINT#0 CHR$(&H1D)+"[0-7]"+CHR$(&H1C)
130 LOCATE27,10:PRINT#0 CHR$(&H1F)
140 LOCATE24,13:PRINT"C = CLS"
150 LOCATE24,14:PRINT"G = GENERATE"
160 LOCATE24,18:PRINT"CHR$ NO. (1-256)"
170 X=8:Y=8
180 LOCATE X,Y : S$=INPUT$(1) : C=ASC(S$)
190 IFC=&H1E THENY=Y-1
200 IFC=&H1D THENX=X-1
210 IFC>=&H30 AND C<=&H37 GOSUB310
220 IFC=&H1F THENY=Y+1
230 IFC=&H1C THENX=X+1
240 IFS$="C"THEN20
250 IFS$="G"THEN350
260 IF X<8 THEN X=8
270 IF X>15 THEN X=15
280 IF Y<8 THEN Y=8
290 IF Y>15 THEN Y=15
300 COLOR7:GOTO180
310 LOCATEX,Y : BEEP
320 C=C-&H30
330 COLOR C:PRINT""(
340 A%(X-8,Y-8)=C:RETURN
350 S$=INKEY$:IFS$<>""THEN350
360 BEEP:LOCATE25,19:INPUT"INPUT=",CH:BEEP
370 IFCH<0ORCH>255 THEN360
380 WIDTH40:O=0 : CH=CH-1
390 F$=F$+"DEF CHR$("+RIGHT$(STR$(CH),LEN(STR$(CH))-1)+")=HEXCHR
$("+CHR$(&H22)
400 FORT=1TO3
410 FOR S=0 TO 7
420 FOR R=0 TO 7
430 E=2^(7-R)
440 ONT GOSUB540,560,580
450 NEXT
460 D$=RIGHT$("00"+HEX$(D),2)
470 G$=G$+D$:D=0
480 NEXT : NEXT
490 F$=F$+G$+CHR$(&H22)+")"+CHR$(&HD)
500 DEF CHR$(CH)=HEXCHR$(G$)
510 PRINT F$
520 CGEN 1 : PRINT#0 CHR$(CH) : CGEN 0
530 PRINT : END
540 IFA%(R,S)=1ORA%(R,S)=3ORA%(R,S)=5ORA%(R,S)=7THEND=D+E
550 RETURN
560 IFA%(R,S)=2ORA%(R,S)=3ORA%(R,S)=6ORA%(R,S)=7THEND=D+E
570 RETURN
580 IFA%(R,S)>3THEND=D+E
590 RETURN
```

トの使い方は，いずれまたハード特集などで取り上げられるでしょう。

FM音源ボードにあるCTCは，X1turboでは標準装備ですが，これまた便利な機能です。CTCというのは，名前のとおりタイマで，一定の時間間隔でCPUに割り込み識別信号を送ります。X1ではテレビ用タイマといっしょに，割り込みのかけられるインターバルタイマを載せているのですが，こちらは1分単位でしか時間設定できません。それに対しCTCは，4μs単位で律儀に割り込みをかけてくれます。

CTCは，FM音源ボード上ではテンポ管理に使われることが多く，事実，MusicBASICなどでもテンポはCTCでとっています。実例としてリスト6を入力してみてください。BASIC上で何を実行させても，かまわず1秒間隔でBEEP音を鳴らし続けます。マシン語サブルーチンを持っているプログラムを新たに走らせると暴走するかもしれませんが，オールBASICプログラムであれば，電源を落とすまで音は消えません。この律儀さがCTCの真髄でしょう。

周辺チップとサブCPU

X1では，キー入力をサブCPUが管理しています。サブCPUも，CTCのところで説明したのと同じように，Z80CPUに対して割り込みをかけてくれます。すなわち，キーボードからキー入力があったら，Z80CPUにキー入力の内容を知らせてくれるのです。もしこの機能がなかったら，Z80はいちいち処理を中断してキーボードを読みにいかねばなりません。キー入力がなくても，その都度読みにいくので，処理速度が遅くなってしまいます。また，キーデータはこのサブCPUがシリアル変換してくれるので，キーボードと本体をつなぐコードが信号線1本ですむわけです。

このように，CPUの負担を減らすことは，そのままユーザーの負担を減らすことにつながります。この割り込みキー入力のおかげでプログラミングもスマートになったわけです。

サブCPUは，キーボードの管理のほか，テレビのコントロールとタイマの管理も行っています。X1は，テレビ画像へのコンピュータディスプレイのスーパーインポーズを最初に実現させたマシンですが，それだけにパソコン側からのテレビのON/OFF，チャンネル切り換え，ボリュームのUP/DOWN操作もサポートしています。これらの機能は，ビデオデッキの普及とともに他のメーカーのパソコンにも相次いで取り入れられたことをみても，時代に先駆けるものであったことがおわかりでしょう。

テレビコントロールに関しては，リスト7でジョイスティックによるリモコン操作のサンプルを挙げておきます。また，先ほど述べたように，ジョイスティック端子をI/Oポートとみなして赤外線受信機でも取りつければ，X1の旧機種でも無線リモコンにすることができるでしょう。

充実した周辺機器

X1の長生きの要因として，周辺ボードのサポートが充実していたことも見逃せません。ハードの世界は進化が急激で，次々により優れたものが登場してきます。

私のマニアタイプにも，ディスクドライブインタフェイス，FM音源ボード，RS-232C/マウスボード，MIDIボードの4枚が取りつけられています。さらに外部メモリボード，立体画像ボード，カラーイメージボードなど，ほしいものは山ほどあります。漢字ROMボードに外部メモリボードを組み合わせてX1turboの漢字VRAMのようなこともできないかと考えています。

このように，周辺機器の豊富なことはユーザーの夢を広げてくれています。

ここでもクリーンコンピュータ思想が重要だということを強調しておきましょう。ハードが進化するにつれ，ソフトウェアも進化していかなければ新しいハードの機能を十分に発揮させられません。ROM BASICが邪魔しているかぎり，結局パソコン本体そのものを交換するのが手っ取り早いことになるでしょう。

X1の場合は，周辺ボードには対応ソフトが付属してきます。またZ-BASICのようにあらかじめハードの拡張を想定したバージョンアップもあります。

こう考えてみると，X1もまだまだ十分使い切られてはいないように思えますね。

やはりX1にこだわる

どうもX1についての記事になると思い入れの強さが前面に出てしまいますが，それほど魅力あるマシンだということです。特に元祖マニアタイプのユーザーとしては，とてもとてもX1はやめられません。HuBASICでプログラミングするだけでも奥の深さは体験できますが，さらにマシン語の世界に足を踏み入れると，もう逃れられなくなるでしょう。そういう「その筋」のストロングタイプを目指すには，『試験に出るX1』を読み，1ページずつ食べていってください。そして全部食べ終わったらもう1冊手に入れて，また隅から隅まで熟読しましょう。そしてぜひ「その筋」なユーザーを目指してハリキッテください。

リスト6　BEEP音を1秒間隔で鳴らす

```
10 '   SAVE "1:SAMPLE6.BAS
20 CLEAR &HFD00
30 CTC=&H1FA0:'for turbo    X1+OPM -> &H0704
40 MEM$(&H5E,2)=MKI$(&HFD00)
50 MEM$(&HFD00,15)=HEXCHR$("F5C5D5E5F3CDF707E1D1C1F1FBED4D")
60 FOR I=1 TO 3
70  READ D : OUT CTC,D
80 NEXT
90 DATA &H58,&B100111,125
100 FOR I=1 TO 2
110  READ D : OUT CTC+3,D
120 NEXT
130 DATA &B11000111,125
```

リスト7　ジョイスティックリモコン操作

```
10 '  SAVE "1:SAMPLE7.BAS
20 JOY=1
30 FLG=0 : TVPW OFF
40 TRIG=STRIG(JOY) : STK=STICK(JOY)
50 IF TRIG=0 AND STK=0 THEN TFLG=0
60 IF TFLG=1 THEN 120
70 IF TRIG=-1 THEN TFLG=1 : GOSUB 130
80 IF STK=8 THEN TFLG=1 : VOL 1
90 IF STK=2 THEN TFLG=1 : VOL -1
100 IF STK=6 THEN TFLG=1 : CH=(CH MOD 12)+1 : CHANNEL CH
110 IF STK=4 THEN TFLG=1 : CH=((CH+10) MOD 12)+1 : CHANNEL CH
120 GOTO 40
130 FLG=1-FLG
140 IF FLG=1 THEN TVPW ON : CRT 3
150 IF FLG=0 THEN TVPW OFF
160 RETURN 40
```

特集1

PCGの基礎から奥義まで

発動！X-700プロジェクト

X1の最大の武器，それはやはりPCGでしょう。ここでは，PCG本来の使い方から少しはずれて，低解像度の擬似グラフィック画面として使用することを考えてみましょう。X1での新しい可能性を探っていきます。

Kameda Masahiko
亀田 雅彦

最近のパソコンは，やれ高解像度だ，やれ6万色だとやたらと高機能化していますが，果たしてそれが本当の姿なのでしょうか。68000や80386などの強力なCPUパワーを持つマシンならば，それなりの機能も当たり前ですが，そんな機能をパーソナルユースでどこまで使いこなせるかは疑問です。だいいち，そういうマシンはワークステーションと呼ばれてしかるべきでしょう。真のパーソナルコンピュータはどこにいってしまったのでしょう。

なんか訳のわからない問題提起調で始まりましたが，要するに私のいいたいことは，現状のX1などの8ビットマシンの位置づけみたいなもんです。それと，本当の意味でのパーソナルとはなんなのか，です。

ユーザーが完全にアプリケーションユーザーのままにされ，プログラマになれない環境にあれば，それは“パーソナル”ではありません。ファミコンなんか，まさにそれでした。また，やろうと思えばできるけどとても個人では手が出せない，なんていうのもパーソナルとはいいがたいものです。そこで，もはや見捨てられた感のある8ビットマシンが登場します。

息が長いだけあって，さすがにプログラミング環境も整備されているし，マシン語も多くの人が知っているので，実に“パーソナル”です。実務（ワープロやレイトレ，アーケードゲームの移植など）はワークステーションにまかせて，私たちアマチュアプログラマはパーソナルな機械でプログラムを作るほうがよいのではないでしょうか。

MZ-700はなぜ速い?

Oh!X読者の皆さんなら，MZ-700のスペハリはご存じですね。キャラクタ画面を駆使して，相当の高速化を実現していました。あのプログラムにおいては，デザイン性（解像度）を切り捨ててもなお，そのゲーム性というものを大事にしていました。それまでは，X1 turbo ZでX68000を目指してばかりいた私には，大きなショックでした。

多量のキャラクタや画面書き換えの必要なゲームではスプライトなどを持たないかぎり，解像度が律速段階になります。X1のPCGというのはもともとこういったことのために装備された機能ですね。しかし，高精度グラフィックで描かれたキャラクタを動かそうとすることは8ビットCPUにとってはアンバランスなことなのです。それをあえて通せば，G-RAMアクセス1回で全プレーンに書き込みを行ったり，スクロール機能を持たせたりCPUのクロックを上げたりといった特殊化されたハードウェアを必要とするようになります。こういったもの用に作られたゲームを，なにも考えずにそのままX1で実行することはX1の設計思想に反することといえるでしょう。

とにかく，X1にはX1の使い方があり，MZ-700のスペハリの示した「MZ-700の正しい使い方」はMZ-700よりも高いポテンシャルを持ったX1でも参考にすべきものが多いのではないでしょうか。X-700はこういったものに基づいた考え方をX1で実現しようという企画です。

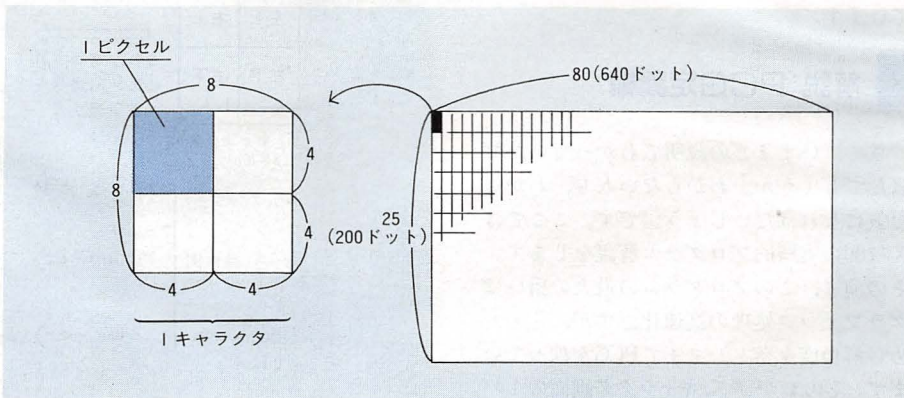
このX-700というプログラムはなんなのかをひと言でいうと，X1のMZ-700化です。誤解を承知のうえであえていうならば，X1の低機能化と高速処理化なのです。低機能化といっても，もちろん基板からRAMを引っこ抜くとかいうことではなく，こんな使い方もあるよという紹介です。その機能が，たまたま時代を逆行するようなものなんですが……。

具体的な説明をしましょう。このX-700システムを使えば，PCGを利用して160×50ピクセル，1ピクセル当たり4色の擬似グラフィック画面を作り出します（もちろん実際のキャラクタ画面上。ちょうどMZ-80Kが，1/4キャラクタで擬似グラフィックを実現していたのと同じようなことをX1においても実現する）。

つまり，実際の1キャラクタ（8×8ドット）を左上・右上・左下・右下の4つの部分（4×4ドット）に分けて考えて，その4×4ドットを1ピクセルとすると，横は640（ドット）÷4で160（ピクセル），縦は200（ドット）÷4で50（ピクセル）になるわけです。このあたりは図1を見てください。

X-700の基本原理

では，その実現方法です。基本的には，PCGにすべてのパターンを定義するわけです。1ピクセル当たり2色（あるか，ないか）なら，左上・右上・左下・右下（1キャラクタ内）にそれぞれあるか，ないかの2通りだから，2×2×2×2で16通りです。1ピクセル当たり4色だから，4×4×4×4でめでたく256通りになるんです。256個と

図1　擬似グラフィック画面の設定

くればPCGの個数とピッタリ一致しますから，都合がいいわけです。どうしても8色(8階調)ほしいという人は80×50の解像度で同じようなことをやってみるといいでしょう。

次に，4ピクセルの組み合わせを，それぞれどのPCGに定義するかが問題になります。ひとつの文字を4分割してどの位置にどの色があるかを調べてPCG番号を決定していたのでは，とても高速処理どころではありません。ここでPCGの数と1キャラクタがとりうるパターンの数が1：1に対応しているということが生きてきます。

まず1バイトを2ビットずつ4つの部分に分解します。座標値からピクセルがこの4つの部分のうちどこに位置するかを求めて，そこにカラーコード（0～3）を書き込むと色の変更ができるようにあらかじめPCGをセットすることもできるわけです。そうすると，これがすなわち，憧れのパックドピクセル，垂直型VRAM構造になってしまうわけですね。

ただ，X68000のように1ドットが1ワードに対応しているわけではありませんから，「なんで垂直型なのにビットマスクがいるの～」ということになってしまいますが，逆に「1回のアクセスで4ドットに書き込みできるので速い」というへ理屈も成り立つわけです。

このX-700では，上位ビットから左上・右上・左下・右下の順に定義してみました。こうしておくと，あとでスクロールなどのとき便利です。ここらへんはなかなか理解しづらいところだと思いますから，図2を見ながら何度も読み返してください。具体的に挙げておくと，00HのPCGは4ピクセルとも0，01Hは右下だけ1であとは0，81Hは左上が2で右下が1であとは0，というふうになっています。

なお，X-700では上記の4色をタイルパターンを使うことによって，パレットとして64色中4色を選ぶことができるようにしています。

解説：PCG定義編

さて，いままでの説明でわかっていただけたでしょうか。わからない人も，わかった気になればだいじょうぶです。ここではX-700の実践的プログラム解説をします。その前に，このプログラムの最大の狙いはグラフィック処理の高速化ですが，そのために解像度を落としてまでPCGを使っています。そのおかげで，キャラクタ画面の1バイトをアクセスするだけで，4ピクセルにアクセスできるようになっています。

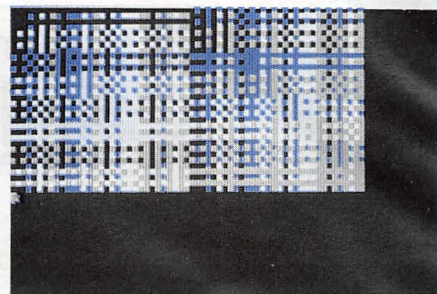
なお，リスト中のアセンブラソースはREDAでアセンブルできます。サンプルのBASICプログラムは，すべてCZ-8FB01で動作するようになっています。

☆PCGセットプログラム：リスト1

しょっぱなのSET44のラベルのつくループが，メインループです。AレジスタにASCIIコードを入れて256回まわしています。その中のSETDTが，PCGに設定するDATAを生成する重要なサブルーチンです。SETPCGはその名のとおり，CRTCの隙を狙ってガシガシと定義するルーチンです。ちなみにこれは，『試験に出るX1』(祝一平著）の中の3倍速定義プログラムとほぼ同じです。この場を借りて，あらためて祝氏に感謝しましょう。

そして，SETDTです。まず，EレジスタにAレジスタのASCIIコードを入れ，RLCECというルーチンに行きます。ここで，Eレジスタの上位2ビット（左上）をもとにPPALETからカラーコードを持ってきて，Aレジスタに入れて戻ります。これを2回やってそれぞれのカラーコードをB，Cレジスタに入れておきます。つまり，ここまでで左上のカラーコードがB，右上のカラーコードがCレジスタに入るわけです。

次にDCDDT4ですが，これはPCG8段飾りのうちの1段を決定するルーチンです。裏BCレジスタに左上，裏DEレジスタに右上のDATAを入れて，B，Cのカラーコードをもとに，裏レジスタのパターンをAレジスタに取り込んでいきます。3回ループしているのはPCGがBRGの3色で構成されているからです。

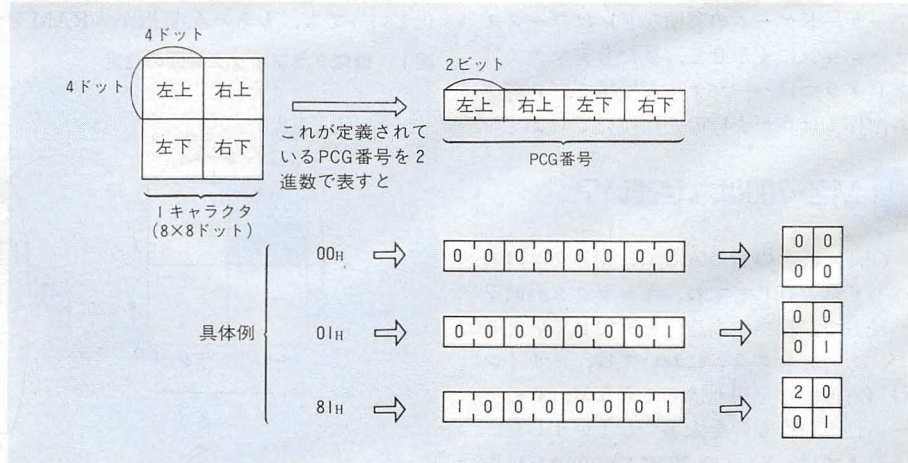
注意してほしいのは，カラーコードが上位4ビットと下位4ビットで構成されながら，その内の3ビットしか参照されていない点です。そもそもカラーコードは，下位ビットから青，赤，緑，関係なしの4ビットで8色を表しています。したがって，1回RRC Bとすれば青のビットが，5回やれば上位4ビットの青が取り出せる仕組みです。

そして，2回DCDDT4をコールしてできた6バイトのデータをコピーして12バイト作ります。今度は左下と右下の12バイトを作るためにDCDDT8をループします。合わせて24バイトのデータを，SETPCGで定義してやっと1個のPCGが完成します。これを256個について行うわけです。

念のため，図3に仕組みを載せておきます。わからない人も，使用上問題ありませんから安心してください。

サンプルプログラムはリスト2です。必ずマシン語部分を&HD000から読み込んでおいてください。カラーパレットを4色聞いてきますから，1バイトの16進数で入力してください。PPALET（カラーパレット情報）は，&HD131から4バイトであること以外特筆すべきことはありません。なお，このSET44ルーチンは，WIDTH80のノーマルX1モードでのみ動作するように設計されています。

忘れるところでしたが，&HD100からこっそりとCLSプログラムが隠されています。

PCGの定義例

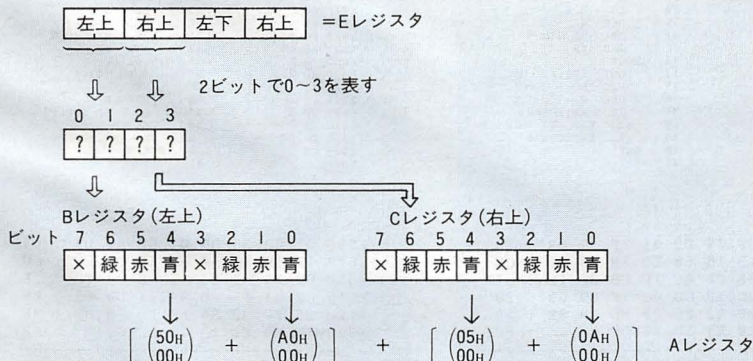
図2　文字コードとの対応

リスト1 PCGセット

```
0000                    1  ;
0000                    2  ;PROGRAM SET PCG
0000                    3  ;
0000                    4
D000                    5   ORG $D000
D000                    6
D000                    7  SET44
D000 01 D0 1F           8   LD   BC,$1FD0
D003 AF                 9   XOR  A
D004 ED 79             10   OUT  (C),A
D006 3E 00             11   LD   A,0
D008                   12  SETLP1
D008 F5                13    PUSH AF
D009 CD 16 D0          14    CALL SETDT
D00C F1                15    POP  AF
D00D F5                16    PUSH AF
D00E CD 98 D0          17    CALL SETPCG
D011 F1                18    POP  AF
D012 3C                19    INC  A
D013 20 F3             20    JR   NZ,SETLP1
D015 C9                21   RET
D016                   22
D016                   23  ;
D016                   24  ;SET PCG DATA
D016                   25  ;
D016                   26  SETDT
D016 21 16 D1          27   LD   HL,WKPCG
D019 5F                28   LD   E,A
D01A 16 02             29   LD   D,2
D01C                   30  DCDDT8
D01C CD 84 D0          31    CALL RLCEC
D01F 47                32    LD   B,A
D020 CD 84 D0          33    CALL RLCEC
D023 4F                34    LD   C,A
D024 D5                35    PUSH DE
D025 E5                36     PUSH HL
D026 D9                37     EXX
D027 06 A0             38     LD   B,$A0
D029 0E 50             39     LD   C,$50
D02B 16 0A             40     LD   D,$0A
D02D 1E 05             41     LD   E,$05
D02F D9                42     EXX
D030 CD 4D D0          43     CALL DCDDT4
D033 D9                44     EXX
D034 06 50             45     LD   B,$50
D036 0E A0             46     LD   C,$A0
D038 16 05             47     LD   D,$05
D03A 1E 0A             48     LD   E,$0A
D03C D9                49     EXX
D03D CD 4D D0          50     CALL DCDDT4
D040 D1                51     POP  DE
D041 EB                52     EX   DE,HL
D042 01 06 00          53     LD   BC,6
D045 ED B0             54     LDIR
D047 EB                55     EX   DE,HL
D048 D1                56    POP  DE
D049 15                57    DEC  D
D04A 20 D0             58    JR   NZ,DCDDT8
D04C C9                59   RET
D04D                   60
D04D                   61  ;
D04D                   62  DCDDT4
D04D C5                63   PUSH BC
D04E 16 03             64   LD   D,3
D050                   65  CLRLP1
D050 AF                66    XOR  A
D051 CB 08             67    RRC  B
D053 30 03             68    JR   NC,NCLR1
D055 D9                69    EXX
D056 B0                70    OR   B
D057 D9                71    EXX
D058                   72  NCLR1
D058 58                73    LD   E,B
D059 CB 08             74    RRC  B
D05B CB 08             75    RRC  B
D05D CB 08             76    RRC  B
D05F CB 08             77    RRC  B
D061 30 03             78    JR   NC,NCLR2
D063 D9                79    EXX
D064 B1                80    OR   C
D065 D9                81    EXX
D066                   82  NCLR2
D066 43                83    LD   B,E
D067 CB 09             84    RRC  C
D069 30 03             85    JR   NC,NCLR3
D06B D9                86    EXX
D06C B2                87    OR   D
D06D D9                88    EXX
D06E                   89  NCLR3
D06E 59                90    LD   E,C
D06F CB 09             91    RRC  C
D071 CB 09             92    RRC  C
D073 CB 09             93    RRC  C
D075 CB 09             94    RRC  C
D077 30 03             95    JR   NC,NCLR4
D079 D9                96    EXX
D07A B3                97    OR   E
D07B D9                98    EXX
D07C                   99  NCLR4
D07C 4B               100    LD   C,E
D07D 77               101    LD   (HL),A
D07E 23               102    INC  HL
D07F 15               103    DEC  D
D080 20 CE            104    JR   NZ,CLRLP1
D082 C1               105   POP  BC
D083 C9               106   RET
D084                  107
D084                  108  ;
D084                  109  RLCEC
D084 C5               110   PUSH BC
D085 E5               111   PUSH HL
D086 CB 03            112   RLC  E
D088 CB 03            113   RLC  E
D08A 7B               114   LD   A,E
D08B E6 03            115   AND  3
D08D 4F               116   LD   C,A
D08E 06 00            117   LD   B,0
D090 21 31 D1         118   LD   HL,PPALET
D093 09               119   ADD  HL,BC
D094 7E               120   LD   A,(HL)
D095 E1               121   POP  HL
D096 C1               122   POP  BC
D097 C9               123   RET
D098                  124
D098                  125  ;
D098                  126  ;SET PCG
D098                  127  ; IN A=CHR.CODE,(WKPCG)
D098                  128  SETPCG
D098 21 00 38         129   LD   HL,$3800
D09B 16 00            130   LD   D,0
D09D CD E7 D0         131   CALL SETP1
D0A0 21 00 20         132   LD   HL,$2000
D0A3 16 20            133   LD   D,$20
D0A5 CD E7 D0         134   CALL SETP1
D0A8 21 00 30         135   LD   HL,$3000
D0AB 57               136   LD   D,A
D0AC CD E7 D0         137   CALL SETP1
D0AF 06 16            138   LD   B,$16
D0B1 0E 00            139   LD   C,0
D0B3 16 17            140   LD   D,$17
D0B5 1E 18            141   LD   E,$18
D0B7 3E 08            142   LD   A,8
D0B9 21 16 D1         143   LD   HL,WKPCG
D0BC                  144
D0BC 08               145   EX   AF,AF'
D0BD D9               146   EXX
D0BE F3               147   DI
D0BF 01 01 1A         148   LD   BC,$1A01
D0C2                  149  VDSP0
D0C2 ED 78            150    IN   A,(C)
D0C4 F2 C2 D0         151    JP   P,VDSP0
D0C7                  152  VDSP1
D0C7 ED 78            153    IN   A,(C)
D0C9 FA C7 D0         154    JP   M,VDSP1
D0CC D9               155   EXX
D0CD 08               156   EX   AF,AF'
D0CE                  157
D0CE                  158  PPPLP
D0CE ED A3            159   OUTI
D0D0 42               160   LD   B,D
D0D1 ED A3            161   OUTI
D0D3 43               162   LD   B,E
D0D4 ED A3            163   OUTI
D0D6 06 16            164   LD   B,$16
D0D8                  165
D0D8 08               166   EX   AF,AF'
D0D9 3E 0B            167   LD   A,$B
D0DB                  168  DLY
D0DB 3D               169   DEC  A
D0DC C2 DB D0         170   JP   NZ,DLY
D0DF 08               171   EX   AF,AF'
D0E0                  172
D0E0 0C               173   INC  C
D0E1 3D               174   DEC  A
D0E2 C2 CE D0         175   JP   NZ,PPPLP
D0E5 FB               176   EI
D0E6 C9               177   RET
D0E7                  178
D0E7                  179  ;
D0E7                  180  SETP1
D0E7 F5               181   PUSH AF
D0E8 C5               182   PUSH BC
D0E9 ED 4B 2E D1      183   LD   BC,(WWITH)
D0ED 09               184   ADD  HL,BC
D0EE 44               185   LD   B,H
D0EF 4D               186   LD   C,L
D0F0 3A 30 D1         187   LD   A,(WWITH+2)
D0F3                  188  SETPLP
D0F3 ED 51            189    OUT  (C),D
D0F5 03               190    INC  BC
D0F6 3D               191    DEC  A
D0F7 20 FA            192    JR   NZ,SETPLP
D0F9 C1               193   POP  BC
D0FA F1               194   POP  AF
D0FB C9               195   RET
D0FC                  196
D0FC                  197  ;
D0FC                  198  ;PROGRAM CLS
D0FC                  199  ;
D0FC                  200
D100                  201   ORG $D100
D100                  202
D100                  203  CLS
D100 01 00 30         204   LD   BC,$3000
D103                  205  CLSLP0
D103 3E 00            206    LD   A,0
D105 ED 79            207    OUT  (C),A
D107 CB A0            208    RES  4,B
D109 3E 27            209    LD   A,$27
D10B ED 79            210    OUT  (C),A
D10D CB E0            211    SET  4,B
D10F 03               212    INC  BC
D110 78               213    LD   A,B
D111 FE 38            214    CP   $38
D113 20 EE            215    JR   NZ,CLSLP0
D115 C9               216   RET
D116                  217
D116                  218  ;
D116                  219  ;DATA AREA
D116                  220  ;
D116                  221  WKPCG
D116                  222   DS   24
D12E                  223
D12E                  224  WWITH
D12E D0 07            225   DW   $7D0 ;WIDTH40=$3E8
D130 30               226   DB   48   ;        =24
D131                  227
D131                  228  PPALET
D131 00 11 22 77      229   DB   0,$11,$22,$77
D135                  230
```

```
D000 01 D0 1F AF ED 79 3E 00 : 43
D008 F5 CD 16 D0 F1 F5 CD 98 : F3
D010 D0 F1 3C 20 F3 C9 21 16 : 10
D018 D1 5F 16 02 CD 84 D0 47 : B0
D020 CD 84 D0 4F D5 E5 D9 06 : 09
D028 A0 0E 50 16 0A 1E 05 D9 : 1A
D030 CD 4D D0 D9 06 50 0E A0 : C7
D038 16 05 1E 0A D9 CD 4D D0 : 06
D040 D1 EB 01 06 00 ED B0 EB : 4B
D048 D1 15 20 D0 C9 C5 16 03 : 7D
D050 AF CB 08 30 03 D9 B0 D9 : 17
D058 58 CB 08 CB 08 CB 08 CB : 9C
D060 08 30 03 D9 B1 D9 43 CB : AC
D068 09 30 03 D9 B2 D9 59 CB : C4
D070 09 CB 09 CB 09 CB 09 30 : B5
D078 03 D9 B3 D9 4B 77 23 15 : 62
-----------------------------------
SUM: AD 6B 88 10 E7 25 7B B1 D37B

D080 20 CE C1 C9 C5 E5 CB 03 : F0
D088 CB 03 7B E6 03 4F 06 00 : 87
D090 21 31 D1 09 7E E1 C1 C9 : 15
D098 21 00 38 16 00 CD E7 D0 : F3
D0A0 21 00 20 16 20 CD E7 D0 : FB
D0A8 21 00 30 57 CD E7 D0 06 : 32
D0B0 16 0E 00 16 17 1E 18 3E : C5
D0B8 08 21 16 D1 08 D9 F3 01 : E5
D0C0 01 1A ED 78 F2 C2 D0 ED : F1
D0C8 78 FA C7 D0 D9 08 ED A3 : 7A
D0D0 42 ED A3 43 ED A3 06 16 : C1
D0D8 08 3E 0B 3D C2 DB D0 08 : 03
D0E0 0C 3D C2 CE D0 FB C9 F5 : 62
D0E8 C5 ED 4B 2E D1 09 44 4D : 96
D0F0 3A 30 D1 ED 51 03 3D 20 : D9
D0F8 FA C1 F1 C9 39 00 39 00 : E7
-----------------------------------
SUM: 55 8B DC 9C F7 DC 51 C1 F10C

D100 01 00 30 3E 00 ED 79 CB : A0
D108 A0 3E 27 ED 79 CB E0 03 : 19
D110 78 FE 38 20 EE C9 39 00 : BE
D118 39 00 39 00 39 00 39 00 : E4
D120 39 00 39 00 39 00 39 00 : E4
D128 39 00 39 00 39 00 D0 07 : 82
D130 30 00 11 22 77          : DA
-----------------------------------
SUM: F4 3C 4B 6D 89 81 D4 D5 1808
```

図3 PCG定義の様子

リスト2 PCG定義サンプル

```
1000 '
1010 'SAMPLE SET PCG
1020 '
1030 WIDTH 80:INIT:CLEAR &HD000:DEFINT A-Z
1040 '
1050 'LOADM "0:SET44.OBJ",&HD000
1060 PADR=&HD131
1070 FOR I=0 TO 3
1080  PRINT I;"=";:INPUT "&H",CW$
1090  POKE PADR+I,VAL("&H"+CW$)
1100 NEXT
1110 CALL &HD000:CALL &HD100
1120 '
1130 W=0
1140 FOR I=0 TO 7
1150  FOR J=0 TO 31
1160   CSIZE3:CGEN1:LOCATE J*2,I*2
1170   PRINT#0 CHR$(W):CGEN0:CSIZE0
1180   W=W+1
1190  NEXT
1200 NEXT
```

アトリビュートVRAMを全部PCGモードにしてくれるので，これからもお世話になるでしょう。

応用：運用編

一度PCGにセットしちゃえばこっちのもんです。あとはキャラクタコードを変えるだけだから，楽ちん楽ちん。

☆PSETプログラム：リスト3

BASICから呼び出せるように，(DE) からX，Y，カラーコード（0～3）を，それぞれ2バイトずつ入れてコールするようになっています(使うのは1バイトのみ)。XYから，カラーコード2ビットが1キャラクタ中のどこにくるのかを判断します。忘れちゃいけないのが，変更する2ビット以外の6ビットを変えないようにマスクしてやることです。そのためのパターンをDレジスタに用意しています。また，後ろのほうでやっているHLの加算は，アドレス算出の常套手段です。

サンプルプログラムはリスト4です。まったくタコですから，ランダムに100個の点を打つだけです。まあ，あとでも使うからよしとしましょう。このサンプルも，マシン語のSET44を&HD000から読み込んで実行したうえで，マシン語のPSETを&HE000から読み込んでおいてください。PSETはリロケータブルプログラムですので，適当なアドレスでいいのですが，サンプル中では&HE000からになっています。

☆スクロールプログラム：リスト5～8

4ドット単位スクロールです（1ピクセル4ドットだから当たり前)。ただその方法が，コロンブスの卵のようになかなか気づかないものなのです。少なくとも私は他人からの入れ知恵でした。

まず，下方向のスクロールです。基本的には，裏と表のBCレジスタを取っ換え引っ換えしながらデータを転送するというX1ではごく普通の方法です。しかし4ドット単位ですから，PCG1個の上半分と下半分を切り離すことになります。そこでこのX-700の本領が発揮されます！　すなわちPCG1個の下半分とは，そのPCGコード8ビットの下位4ビットで，上半分は上位4ビットです。だから上のPCGの下位4ビットと下の上位4ビットを合わせてOUTするだけでスクロールするのです。なんかぐちゃぐちゃでわからないかもしれませんが，図4をみてもらえば一目瞭然でしょう。上方向

リスト3 PSET

```
0000                1  ;
0000                2  ;PROGRAM PSET
0000                3  ; IN (DE+0..4)=X,Y,COLOR
0000                4
E000                5   ORG $E000
E000                6
E000                7  PSET
E000 EB             8   EX   DE,HL
E001 46             9   LD   B,(HL)
E002 23            10   INC  HL
E003 23            11   INC  HL
E004 4E            12   LD   C,(HL)
E005 23            13   INC  HL
E006 23            14   INC  HL
E007 7E            15   LD   A,(HL)
E008 E6 03         16   AND  3
E00A 0F            17   RRCA
E00B 0F            18   RRCA
E00C 57            19   LD   D,A   ;D=&B??000000
E00D 3E 3F         20   LD   A,$3F ;A=&B00111111
E00F B7            21   OR   A
E010 CB 18         22   RR   B
E012 30 06         23   JR   NC,NXRGHT
E014 0F            24    RRCA
E015 0F            25    RRCA
E016 CB 0A         26    RRC  D
E018 CB 0A         27    RRC  D
E01A               28  NXRGHT
E01A B7            29   OR   A
E01B CB 19         30   RR   C
E01D 30 0C         31   JR   NC,NYUNDR
E01F 0F            32    RRCA
E020 0F            33    RRCA
E021 0F            34    RRCA
E022 0F            35    RRCA
E023 CB 0A         36    RRC  D
E025 CB 0A         37    RRC  D
E027 CB 0A         38    RRC  D
E029 CB 0A         39    RRC  D
E02B               40  NYUNDR
E02B 69            41   LD   L,C
E02C 26 00         42   LD   H,0
E02E 48            43   LD   C,B
E02F 44            44   LD   B,H
E030 29            45   ADD  HL,HL
E031 29            46   ADD  HL,HL
E032 29            47   ADD  HL,HL
E033 29            48   ADD  HL,HL ;WIDTH40..NOP
E034 E5            49   PUSH HL
E035 29            50   ADD  HL,HL
E036 29            51   ADD  HL,HL
E037 09            52   ADD  HL,BC
E038 C1            53   POP  BC
E039 09            54   ADD  HL,BC
E03A 01 00 30      55   LD   BC,$3000
E03D 09            56   ADD  HL,BC
E03E 44            57   LD   B,H
E03F 4D            58   LD   C,L
E040 ED 58         59   IN   E,(C)
E042 A3            60   AND  E
E043 B2            61   OR   D
E044 ED 79         62   OUT  (C),A
E046 C9            63   RET
E047               64

E000 EB 46 23 23 4E 23 23 7E : 89
E008 E6 03 0F 0F 57 3E 3F B7 : 92
E010 CB 18 30 06 0F 0F CB 0A : 0C
E018 CB 0A B7 CB 19 30 0C 0F : BB
E020 0F 0F 0F CB 0A CB 0A CB : A2
E028 0A CB 0A 69 26 00 48 44 : FA
E030 29 29 29 29 E5 29 29 09 : E4
E038 C1 09 01 00 30 09 44 4D : 95
E040 ED 58 A3 B2 ED 79 C9    : C9
---------------------------------
SUM: 57 CF FF 12 FF 16 C1 B3 75AF
```

リスト4 PSETサンプル

```
1000 '
1010 'SAMPLE PSET
1020 '
1030 WIDTH 80:INIT:CLEAR &HD000:DEFINT A-Z
1040 ADR=&HE000
1050 'LOADM "0:SET44.OBJ",&HD000
1060 'LOADM "0:PSET.OBJ",ADR
1070 'CALL &HD000
1080 '
1090 CALL &HD100:DEFUSR0=ADR
1100 FOR I=1 TO 100
1110   X1=INT(RND*160):Y1=INT(RND*50):CL=INT(RND*4)
1120   A$=USR0(MKI$(X1)+MKI$(Y1)+MKI$(CL))
1130 NEXT
```

リスト5 SCROLL DOWN

```
0000                1  ;
0000                2  ;PROGRAM SCROLL DOWN
0000                3  ;
0000                4
E000                5   ORG $E000
E000                6
E000                7  SCROLD
E000 01 CF 37       8   LD   BC,$3000+1999
E003 D9             9   EXX
E004 01 7F 37      10   LD   BC,$3000+1919
E007 D9            11   EXX
E008 1E 19         12   LD   E,25
E00A               13  SCRLP0
E00A 16 50         14    LD   D,80
E00C               15  SCRLP1
E00C ED 78         16      IN   A,(C)
E00E D9            17      EXX
E00F ED 60         18      IN   H,(C)
E011 0B            19      DEC  BC
E012 CB 0C         20      RRC  H
E014 1F            21      RRA
E015 CB 0C         22      RRC  H
E017 1F            23      RRA
E018 CB 0C         24      RRC  H
E01A 1F            25      RRA
E01B CB 0C         26      RRC  H
E01D 1F            27      RRA
E01E D9            28      EXX
E01F 63            29      LD   H,E
E020 25            30      DEC  H
E021 20 02         31      JR   NZ,GYOU0
E023 E6 0F         32      AND  $F
E025               33  GYOU0
E025 ED 79         34      OUT  (C),A
E027 0B            35      DEC  BC
E028 15            36      DEC  D
E029 20 E1         37      JR   NZ,SCRLP1
E02B 1D            38    DEC  E
E02C 20 DC         39    JR   NZ,SCRLP0
E02E C9            40   RET
E02F               41

E000 01 CF 37 D9 01 7F 37 D9 : 70
E008 1E 19 16 50 ED 78 D9 ED : C8
E010 60 0B CB 0C 1F CB 0C 1F : 57
E018 CB 0C 1F CB 0C 1F D9 63 : 28
E020 25 20 02 E6 0F ED 79 0B : AD
E028 15 20 E1 1D 20 DC C9    : F8
---------------------------------
SUM: 84 3F 1A 03 48 AA 37 53 055F
```

リスト6 SCROLL UP

```
0000                1  ;
0000                2  ;PROGRAM SCROLL UP
0000                3  ;
0000                4
E000                5   ORG $E000
E000                6
E000                7  SCROLD
E000 01 00 30       8   LD   BC,$3000
E003 D9             9   EXX
E004 01 50 30      10   LD   BC,$3000+80
E007 D9            11   EXX
E008 1E 19         12   LD   E,25
E00A               13  SCRLP0
E00A 16 50         14    LD   D,80
E00C               15  SCRLP1
E00C ED 78         16      IN   A,(C)
E00E D9            17      EXX
E00F ED 60         18      IN   H,(C)
E011 03            19      INC  BC
E012 CB 04         20      RLC  H
E014 17            21      RLA
E015 CB 04         22      RLC  H
E017 17            23      RLA
E018 CB 04         24      RLC  H
E01A 17            25      RLA
E01B CB 04         26      RLC  H
E01D 17            27      RLA
E01E D9            28      EXX
E01F 63            29      LD   H,E
E020 25            30      DEC  H
E021 20 02         31      JR   NZ,GYOU0
E023 E6 F0         32      AND  $F0
E025               33  GYOU0
E025 ED 79         34      OUT  (C),A
E027 03            35      INC  BC
E028 15            36      DEC  D
E029 20 E1         37      JR   NZ,SCRLP1
E02B 1D            38    DEC  E
E02C 20 DC         39    JR   NZ,SCRLP0
E02E C9            40   RET
E02F               41

E000 01 00 30 D9 01 50 30 D9 : 64
E008 1E 19 16 50 ED 78 D9 ED : C8
E010 60 03 CB 04 17 CB 04 17 : 2F
E018 CB 04 17 CB 04 17 D9 63 : 08
E020 25 20 02 E6 F0 ED 79 03 : 86
E028 15 20 E1 1D 20 DC C9    : F8
---------------------------------
SUM: 84 60 0B FB 19 73 28 43 FFD6
```

のスクロールもまったく同様です。

左右方向ですが，これも左半分と右半分を分けて考えます。今度は2ビットおきになってしまいますが，それをずらして隣のPCGと合わせるという考え方は一緒です。こんなことでできるのか？　と一瞬思いますがよく考えると当たり前なんですね。PCGはほんとに奥が深いです。

サンプルはリスト9です。例によってSET 44を読み込んで実行したあと，PSETを&HE000から，スクロールプログラムは上下左右どれかひとつを&HD200から読み込んでおいてください。スクロールもすべてリロケータブルです。このPSETの部分は前のPSETサンプルを拝借しています。なんとタコなことに，一度消えたものは戻ってきません。スクロールプログラムのせいなので皆さんビシバシ改造してください。

☆ラインプログラム：リスト10

いわずとしれたラインです。BASICからのコールを考慮に入れて，(DE)からX0, Y0, X1, Y1, カラーコード（0〜3）をそれぞれ2バイトずつ入れておきます。実はこのラインプログラムは7月号36ページの丹氏と同じBresenhamのアルゴリズムを使っています。詳しい説明はそちらに譲るとして，ここではX1でどのように実現しているのかを解説しましょう。7月号を手元においてください。

最初のひと区切り，注釈のDE＝2＊ABS（X1−X0）の行まではSX（SGN（X1−X0)）と，DX（ABS（X1−X0)）と，注釈のDEを用意するためのものです。また，注釈のBC＝2＊ABS（Y1−Y0）の行までもXがYに変わっただけです。そうしておいて裏レジスタに用意しておいたものを入れるのですが，これは高速化のためのもので，そのままIXを使っても構いません。

リスト7 SCROLL RIGHT

```
0000                 1  ;
0000                 2  ;PROGRAM SCROLL RIGHT
0000                 3  ;
0000                 4
E000                 5   ORG $E000
E000                 6
E000                 7  SCROLR
E000 01 CF 37        8   LD   BC,$3000+1999
E003 D9              9   EXX
E004 01 CE 37       10   LD   BC,$3000+1998
E007 D9             11   EXX
E008 1E 19          12   LD   E,25
E00A                13  SCRLP0
E00A 16 50          14    LD   D,80
E00C                15  SCRLP1
E00C ED 78          16      IN   A,(C)
E00E 0F             17      RRCA
E00F 0F             18      RRCA
E010 E6 33          19      AND  $33
E012 D9             20      EXX
E013 6F             21      LD   L,A
E014 ED 60          22      IN   H,(C)
E016 0B             23      DEC  BC
E017 CB 04          24      RLC  H
E019 CB 04          25      RLC  H
E01B 7C             26      LD   A,H
E01C E6 CC          27      AND  $CC
E01E B5             28      OR   L
E01F D9             29      EXX
E020 62             30      LD   H,D
E021 25             31      DEC  H
E022 20 02          32      JR   NZ,GYOU0
E024 E6 33          33      AND  $33
E026                34  GYOU0
E026 ED 79          35      OUT  (C),A
E028 0B             36      DEC  BC
E029 15             37      DEC  D
E02A 20 E0          38      JR   NZ,SCRLP1
E02C 1D             39     DEC  E
E02D 20 DB          40     JR   NZ,SCRLP0
E02F C9             41   RET
E030                42


E000 01 CF 37 D9 01 CE 37 D9 : BF
E008 1E 19 16 50 ED 78 0F 0F : 20
E010 E6 33 D9 6F ED 60 0B CB : 84
E018 04 CB 04 7C E6 CC B5 D9 : 8F
E020 62 25 20 02 E6 33 ED 79 : 28
E028 0B 15 20 E0 1D 20 DB C9 : 01
---------------------------------
SUM: 76 20 6A F6 C4 C5 CE CE 9E93
```

リスト8 SCROLL LEFT

```
0000                 1  ;
0000                 2  ;PROGRAM SCROLL LEFT
0000                 3  ;
0000                 4
E000                 5   ORG $E000
E000                 6
E000                 7  SCROLL
E000 01 00 30        8   LD   BC,$3000
E003 D9              9   EXX
E004 01 01 30       10   LD   BC,$3000+1
E007 D9             11   EXX
E008 1E 19          12   LD   E,25
E00A                13  SCRLP0
E00A 16 50          14    LD   D,80
E00C                15  SCRLP1
E00C ED 78          16      IN   A,(C)
E00E 07             17      RLCA
E00F 07             18      RLCA
E010 E6 CC          19      AND  $CC
E012 D9             20      EXX
E013 6F             21      LD   L,A
E014 ED 60          22      IN   H,(C)
E016 03             23      INC  BC
E017 CB 0C          24      RRC  H
E019 CB 0C          25      RRC  H
E01B 7C             26      LD   A,H
E01C E6 33          27      AND  $33
E01E B5             28      OR   L
E01F D9             29      EXX
E020 62             30      LD   H,D
E021 25             31      DEC  H
E022 20 02          32      JR   NZ,GYOU0
E024 E6 CC          33      AND  $CC
E026                34  GYOU0
E026 ED 79          35      OUT  (C),A
E028 03             36      INC  BC
E029 15             37      DEC  D
E02A 20 E0          38      JR   NZ,SCRLP1
E02C 1D             39     DEC  E
E02D 20 DB          40     JR   NZ,SCRLP0
E02F C9             41   RET
E030                42


E000 01 00 30 D9 01 01 30 D9 : 15
E008 1E 19 16 50 ED 78 07 07 : 10
E010 E6 CC D9 6F ED 60 03 CB : 15
E018 0C CB 0C 7C E6 33 B5 D9 : 06
E020 62 25 20 02 E6 CC ED 79 : C1
E028 03 15 20 E0 1D 20 DB C9 : F9
---------------------------------
SUM: 76 EA 6B F6 C4 F8 B7 C6 0D90
```

図4　概念的スクロールの動作状況

リスト9 SCROLLサンプル

```
1000 '
1010 'SAMPLE SCROLL ALL
1020 '
1030 WIDTH 80:INIT:CLEAR &HD000:DEFINT A-Z
1040 ADR=&HE000
1050 'LOADM "0:SET44.OBJ",&HD000
1060 'LOADM "0:PSET.OBJ",&HE000
1070 'CALL &HD000
1080 '
1090 CALL &HD100:DEFUSR0=ADR
1100 FOR I=1 TO 100
1110  X1=INT(RND*160):Y1=INT(RND*50):CL=INT(RND*4)
1120  A$=USR0(MKI$(X1)+MKI$(Y1)+MKI$(CL))
1130 NEXT
1140 '
1150 'LOADM "0:SCROLD.OBJ",&HD200
1160 '
1170 FOR I=1 TO 50
1180  CALL &HD200
1190 NEXT
```

図5　ミクロに見たスクロールの実際

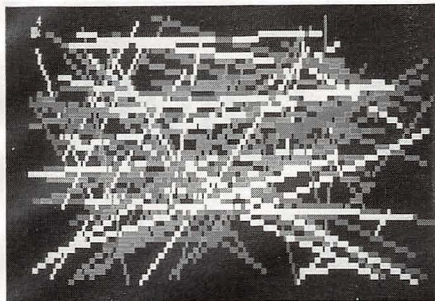
この次に，ラインの傾きによって処理ルーチンを分けます。そしてHLには誤差(負)が入ります。ここからはほぼアルゴリズムどおりですが，誤差(HL)に足したり引いたりしたあとのキャリフラグの条件付きジャンプは，正→負・負→正どちらの場合にも使われていることが特殊です。PSET0ルーチンは前のPSETをちょっと変更したもので，裏B，CレジスタにX，Yを入れてコールします。

サンプルはリスト11です。SET44を&HD000からおいて実行して，LINEプログラムを&HE000から読み込んでください。ちなみにこれはリロケータブルではありません。少し速さも気になったのでTIME命令

リスト10 LINE

```
0000                1  ;
0000                2  ;PROGRAM LINE
0000                3  ; IN (DE+0..8)=X0,Y0,X1,Y1,COLOR
0000                4
E000                5   ORG $E000
E000                6
E000                7  LINE
E000 21 A7 E0       8   LD   HL,WLINEB
E003 01 0A 00       9   LD   BC,10
E006 EB            10   EX   DE,HL
E007 ED B0         11   LDIR
E009 DD 21 A7 E0   12   LD   IX,WLINEB
E00D               13
E00D DD 7E 04      14   LD   A,(IX+4)
E010 DD 5E 00      15   LD   E,(IX+0)
E013 16 01         16   LD   D,1
E015 BB            17   CP   E
E016 30 08         18   JR   NC,SKPLN0
E018 DD 7E 00      19    LD   A,(IX+0)
E01B DD 5E 04      20    LD   E,(IX+4)
E01E 16 FF         21    LD   D,$FF
E020               22  SKPLN0
E020 93            23   SUB  E
E021 DD 72 0A      24   LD   (IX+10),D ;SX
E024 6F            25   LD   L,A
E025 26 00         26   LD   H,0
E027 29            27   ADD  HL,HL
E028 EB            28   EX   DE,HL      ;DE=2*ABS(X1-X0)
E029               29
E029 DD 77 0C      30   LD   (IX+12),A ;DX
E02C               31
E02C DD 7E 06      32   LD   A,(IX+6)
E02F DD 4E 02      33   LD   C,(IX+2)
E032 06 01         34   LD   B,1
E034 B9            35   CP   C
E035 30 08         36   JR   NC,SKPLN1
E037 DD 7E 02      37    LD   A,(IX+2)
E03A DD 4E 06      38    LD   C,(IX+6)
E03D 06 FF         39    LD   B,$FF
E03F               40  SKPLN1
E03F 91            41   SUB  C
E040 DD 70 0B      42   LD   (IX+11),B ;SY
E043 DD 77 0D      43   LD   (IX+13),A ;DY
E046 87            44   ADD  A,A
E047 4F            45   LD   C,A
E048 06 00         46   LD   B,0        ;BC=2*ABS(Y1-Y0)
E04A               47
E04A D9            48   EXX
E04B DD 46 00      49   LD   B,(IX+0)
E04E DD 4E 02      50   LD   C,(IX+2)
E051 DD 56 0A      51   LD   D,(IX+10) ;SX
E054 DD 5E 0B      52   LD   E,(IX+11) ;SY
E057 DD 66 0C      53   LD   H,(IX+12) ;DX
E05A DD 6E 0D      54   LD   L,(IX+13) ;DY
E05D 7C            55   LD   A,H
E05E BD            56   CP   L
E05F 38 23         57   JR   C,LINE2    ;DX-DY?
E061               58
E061 7C            59   LD   A,H
E062 24            60   INC  H
E063 D9            61   EXX
E064 ED 44         62   NEG
E066 6F            63   LD   L,A
E067 26 FF         64   LD   H,$FF      ;HL=ERR
E069               65  LINLP0
E069 CD B5 E0      66    CALL PSET0
E06C D9            67    EXX
E06D 78            68    LD   A,B
E06E 82            69    ADD  A,D
E06F 47            70    LD   B,A
E070 D9            71    EXX
E071 09            72    ADD  HL,BC     ;ERR=ERR+2(Y1-Y0)
E072 30 0A         73    JR   NC,LINLP1
E074               74  LINLP2
E074 D9            75     EXX
E075 79            76     LD   A,C
E076 83            77     ADD  A,E
E077 4F            78     LD   C,A
E078 D9            79     EXX
E079 B7            80     OR   A
E07A ED 52         81     SBC  HL,DE    ;ERR=ERR-2(X1-X0)
E07C 30 F6         82     JR   NC,LINLP2
E07E               83  LINLP1
E07E D9            84    EXX
E07F 25            85    DEC  H
E080 D9            86    EXX
E081 20 E6         87    JR   NZ,LINLP0
E083 C9            88   RET
E084               89
E084               90  LINE2
E084 7D            91   LD   A,L
E085 2C            92   INC  L
E086 D9            93   EXX
E087 ED 44         94   NEG
E089 6F            95   LD   L,A
E08A 26 FF         96   LD   H,$FF      ;HL=ERR
E08C               97  LINLP3
E08C CD B5 E0      98    CALL PSET0
E08F D9            99    EXX
E090 79           100    LD   A,C
E091 83           101    ADD  A,E
E092 4F           102    LD   C,A
E093 D9           103    EXX
E094 19           104    ADD  HL,DE     ;ERR=ERR+2(X0-X1)
E095 30 0A        105    JR   NC,LINLP4
E097              106  LINLP5
E097 D9           107     EXX
E098 78           108     LD   A,B
E099 82           109     ADD  A,D
E09A 47           110     LD   B,A
E09B D9           111     EXX
E09C B7           112     OR   A
E09D ED 42        113     SBC  HL,BC    ;ERR=ERR-2(Y0-Y1)
E09F 30 F6        114     JR   NC,LINLP5
E0A1              115  LINLP4
E0A1 D9           116    EXX
E0A2 2D           117    DEC  L
E0A3 D9           118    EXX
E0A4 20 E6        119    JR   NZ,LINLP3
E0A6 C9           120   RET
E0A7              121
E0A7              122  WLINEB
E0A7 00 00        123  X0    DW 0 ;+0
E0A9 00 00        124  Y0    DW 0 ;+2
E0AB 00 00        125  X1    DW 0 ;+4
E0AD 00 00        126  Y1    DW 0 ;+6
E0AF 00 00        127  COLOR DW 0 ;+8
E0B1 00           128  SX    DB 0 ;+10
E0B2 00           129  SY    DB 0 ;+11
E0B3 00           130  DX    DB 0 ;+12
E0B4 00           131  DY    DB 0 ;+13
E0B5              132
E0B5              133  ;
E0B5              134  ;SUBROUTINE PSET
E0B5              135  ;
E0B5              136  PSET0
E0B5 D9           137   EXX
E0B6 C5           138   PUSH BC
E0B7 D5           139   PUSH DE
E0B8 E5           140   PUSH HL
E0B9 DD 7E 08     141   LD   A,(IX+8)
E0BC E6 03        142   AND  3
E0BE 0F           143   RRCA
E0BF 0F           144   RRCA
E0C0 57           145   LD   D,A   ;D=&B??000000
E0C1 3E 3F        146   LD   A,$3F ;A=&B00111111
E0C3 B7           147   OR   A
E0C4 CB 18        148   RR   B
E0C6 30 06        149   JR   NC,NXRGHT
E0C8 0F           150    RRCA
E0C9 0F           151    RRCA
E0CA CB 0A        152    RRC  D
E0CC CB 0A        153    RRC  D
E0CE              154  NXRGHT
E0CE B7           155   OR   A
E0CF CB 19        156   RR   C
E0D1 30 0C        157   JR   NC,NYUNDR
E0D3 0F           158    RRCA
E0D4 0F           159    RRCA
E0D5 0F           160    RRCA
E0D6 0F           161    RRCA
E0D7 CB 0A        162    RRC  D
E0D9 CB 0A        163    RRC  D
E0DB CB 0A        164    RRC  D
E0DD CB 0A        165    RRC  D
E0DF              166  NYUNDR
E0DF 69           167   LD   L,C
E0E0 26 00        168   LD   H,0
E0E2 48           169   LD   C,B
E0E3 44           170   LD   B,H
E0E4 29           171   ADD  HL,HL
E0E5 29           172   ADD  HL,HL
E0E6 29           173   ADD  HL,HL
E0E7 29           174   ADD  HL,HL ;WIDTH40..NOP
E0E8 E5           175   PUSH HL
E0E9 29           176   ADD  HL,HL
E0EA 29           177   ADD  HL,HL
E0EB 09           178   ADD  HL,BC
E0EC C1           179   POP  BC
E0ED 09           180   ADD  HL,BC
E0EE 01 00 30     181   LD   BC,$3000
E0F1 09           182   ADD  HL,BC
E0F2 44           183   LD   B,H
E0F3 4D           184   LD   C,L
E0F4 ED 58        185   IN   E,(C)
E0F6 A3           186   AND  E
E0F7 B2           187   OR   D
E0F8 ED 79        188   OUT  (C),A
E0FA E1           189   POP  HL
E0FB D1           190   POP  DE
E0FC C1           191   POP  BC
E0FD D9           192   EXX
E0FE C9           193   RET
E0FF              194
```

```
E000 21 A7 E0 01 0A 00 EB ED : 8B
E008 B0 DD 21 A7 E0 DD 7E 04 : 94
E010 DD 5E 00 16 01 BB 30 08 : 45
E018 DD 7E 00 DD 5E 04 16 FF : AF
E020 93 DD 72 0A 6F 26 00 29 : AA
E028 EB DD 77 0C DD 7E 06 DD : 89
E030 4E 02 06 01 B9 30 08 DD : 25
E038 7E 02 DD 4E 06 06 FF 91 : 47
E040 DD 70 0B DD 77 0D 87 4F : 8F
E048 06 00 D9 DD 46 00 DD 4E : 2D
E050 02 DD 56 0A DD 5E 0B DD : 62
E058 66 0C DD 6E 0D 7C BD 38 : 3B
E060 23 7C 24 D9 ED 44 6F 26 : 62
E068 FF CD B5 E0 D9 78 82 47 : 7B
E070 D9 09 30 0A D9 79 83 4F : 40
E078 D9 B7 ED 52 30 F6 D9 25 : F3
-----------------------------------
SUM: F4 80 DA 47 CA 88 35 FF 9810

E080 D9 20 E6 C9 7D 2C D9 ED : 17
E088 44 6F 26 FF CD B5 E0 D9 : 13
E090 79 83 4F D9 19 30 0A D9 : 50
E098 78 82 47 D9 B7 ED 42 30 : 30
E0A0 F6 D9 2D D9 20 E6 C9 00 : A4
E0A8 00 00 00 00 00 00 00 00 : 00
E0B0 00 00 00 00 00 D9 C5 D5 : 73
E0B8 E5 DD 7E 08 E6 03 0F 0F : 4F
E0C0 57 3E 3F B7 CB 18 30 06 : A4
E0C8 0F 0F CB 0A CB 0A B7 CB : 4A
E0D0 19 30 0C 0F 0F 0F 0F CB : 5C
E0D8 0A CB 0A CB 0A CB 0A 69 : F2
E0E0 26 00 48 44 29 29 29 29 : 56
E0E8 E5 29 29 09 C1 09 01 00 : 0B
E0F0 30 09 44 4D ED 58 A3 B2 : 64
E0F8 ED 79 E1 D1 C1 D9 C9    : 7B
-----------------------------------
SUM: 9A 3D 03 61 67 1F 38 93 CCD5
```

ライン……です

が埋め込まれています。BASICのLINE命令や空ループの時間と比べてみるのもオツでしょう。

☆まとめのサンプル:リスト12

いままでのプログラムをまとめて使ってみようというサンプルです。しかし，サンプルはあくまでサンプルです。とてもグラフィックエディタと呼べるものではありません。あしからず。

SET44，PSET，4つのスクロールプログラム，LINEをREM文で殺してあるようなアドレスに読み込んで&HD000をコールしておいてください。一度やればリセットをかけるまで有効です。

RUNすると，小さなカーソルが出るのでテンキーで動かしてください。スペースキーを押せばLINEの始点となり，もう1回押せばそこまでLINEが描けます。リターンキーでPSETを行い，ZXCVの各キーでカラーコードの0123を指定します。カーソルキーの上下左右でズリッズリッとスクロールします。サンプルプログラム自体はきわめて単純ですが，思ったとおり遅くなってしまいました。

まとめ:完結編

このX-700はなかなか応用範囲の広いプログラムです。グラフィック画面と割り切っても使えるし，ゲームのための高速キャラクタ画面としても使えます。たとえば3Dグラフィックをやる場合でも，実際のグラフィック画面じゃ遅いときがあります。そんな場合は，解像度や色数を落としても，速いX-700が役に立つでしょう。ゲームを作りたいとします。けどX1はX68000じゃないんだから，アーケードほどの綿密な画像は所詮無理なんです。ならばデザインで人を引きつけるゲームよりも，原点に帰った本当の意味でのゲーム性で勝負してみることこそ本望でしょう。

さらにX-700システムの最大の長所は1画面まるごとでも2Kバイトしか消費しないということです。G-RAM上のデータを4階調に変換しテキストに取り込んでやると，1画面が2Kバイト，これをまたG-RAMに格納すると24枚分のデータになります。これは簡単な転送ルーチンでVRAMに落とせますから，BASICを使わなければ40枚くらいのアニメーションは簡単に行えます。また，データに圧縮をかけてX1turboのG-RAM，拡張メモリを使えばオンメモリで100枚単位のデータを扱うことも可能となります。

4色しか使えないのはイマイチという気もしますが，X1のキャラクタでは簡単に色反転やプレーンマスクができますから，同じパターンでも特殊効果的に画面上の色数を増やすことは可能です。たとえば，01，11，51，57のような青系統の4階調パターンを作成したとしましょう。これで作ったパターンにリバースをかけてやれば76，66，26，20といった赤系統のグラデーションパターンとして使用できます。PCGの再定義はけっこう面倒ですのでこういった手法も活用すべきでしょう。

考え方を示しただけで，けっこうプログラムは貧弱だったかなとも思いましたが，ここから先は読者の皆さんに育ててもらいたいものです。特にゲームなんか作ろうと思っている人なんかにはおいしいと思うんですが……。

最後に哲学めいたことをいわせてもらうと，X1もまだ捨てたもんじゃない！ というよりも，ユーザー自身が使っていくことによってその存在価値を見い出せるのです。市販ソフトの出る出ないよりも，ユーザーがパーソナルに使うかどうかということに意味があると思います。

リスト11 LINEサンプル

```
1000 '
1010 'SAMPLE LINE
1020 '
1030 WIDTH 80:INIT:CLEAR &HD000:DEFINT A-Z
1040 ADR=&HE000
1050 'LOADM "0:SET44.OBJ",&HD000
1060 'LOADM "0:LINE.OBJ",ADR
1070 'CALL &HD000
1080 '
1090 CALL &HD100:DEFUSR0=ADR
1100 TIME=0
1110 FOR I=1 TO 100
1120  X0=INT(RND*160):Y0=INT(RND*50)
1130  X1=INT(RND*160):Y1=INT(RND*50):CL=INT(RND*3+1)
1140  A$=USR0(MKI$(X0)+MKI$(Y0)+MKI$(X1)+MKI$(Y1)+MKI$(CL))
1150 NEXT
1160 PRINT TIME
```

リスト12 簡易エディタ

```
1000 '
1010 'SAMPLE EDITER
1020 '
1030 WIDTH 80:INIT:CLEAR &HD000:DEFINT A-Z:DIM ADRS(3),SX(9),SY(9)
1040 CLICK OFF:PRW &H80
1050 ADR0=&HD200:ADR1=&HE000
1060 ADRS(0)=&HD300:ADRS(1)=&HD400
1070 ADRS(2)=&HD500:ADRS(3)=&HD600
1080 'LOADM "0:SET44.OBJ",&HD000
1090 'LOADM "0:PSET.OBJ",ADR0
1100 'LOADM "0:SCROLD.OBJ",ADRS(3):LOADM "0:SCROLU.OBJ",ADRS(2)
1110 'LOADM "0:SCROLL.OBJ",ADRS(1):LOADM "0:SCROLR.OBJ",ADRS(0)
1120 'LOADM "0:LINE.OBJ",ADR1
1130 'CALL &HD000
1140 '
1150 CALL &HD100:DEFUSR0=ADR0:DEFUSR1=ADR1
1160 MX=0:MY=0:CL=3:F0=0
1170 FOR I=0 TO 9:READ A,B:SX(I)=A:SY(I)=B:NEXT
1180 DATA 0,0,-8,4,0,4,8,4,-8,0,0,0,8,0,-8,-4,0,-4,8,-4
1190 GOSUB 1360
1200 '
1210 GOSUB 1360:IF F0=1 THEN C0= 0:GOSUB 1390
1220 ST=STICK(0):MX=MX+SX(ST):MY=MY+SY(ST)
1230 IF MX<0 OR MX>639 OR MY<0 OR MY>199 THEN MX=MX-SX(ST):MY=MY-SY(ST)
1240 GOSUB 1360:IF F0=1 THEN C0=CL:GOSUB 1390
1250 SR$=INKEY$:IF SR$="" GOTO 1200
1260 IF SR$=" " AND F0=0 THEN X0=MX/4:Y0=MY/4:F0=1:BEEP :GOTO 1200
1270 IF SR$=" " AND F0=1 THEN C0=CL:GOSUB 1390:F0=0:BEEP:GOTO 1200
1280 IF SR$=CHR$(13) THEN A$=USR0(MKI$(MX/4)+MKI$(MY/4)+MKI$(CL)):GOTO 1200
1290 IF SR$=CHR$(12) THEN CALL &HD100:GOTO 1200
1300 IF SR$="Z" THEN CL=0:BEEP:GOTO 1200
1310 IF SR$="X" THEN CL=1:BEEP:GOTO 1200
1320 IF SR$="C" THEN CL=2:BEEP:GOTO 1200
1330 IF SR$="V" THEN CL=3:BEEP:GOTO 1200
1340 IF &H1C<=ASC(SR$) AND ASC(SR$)<=&H1F THEN CALL ADRS(ASC(SR$)-&H1C)
1350 GOTO 1200
1360 '
1370 LINE (MX-1,MY)-(MX+1,MY),XOR,7:LINE (MX,MY-1)-(MX,MY+1),XOR,7
1380 RETURN
1390 '
1400 A$=USR1(MKI$(X0)+MKI$(Y0)+MKI$(MX/4)+MKI$(MY/4)+MKI$(C0))
1410 RETURN
```

特集1

楽々線引きプログラム

高速ラインルーチンG-LINE

Ohno Naoyuki
大野 直之

どちらかというと高速グラフィックは苦手としているX1に，高速ラインルーチンをお届けします。速度はあのMAGICの約2倍。多少の変更で他機種でも使用できますので，移植に挑戦してみるのもよいでしょう。活用してください。

スタンドアロンの線引きプログラムを作ってみました。LINEのスピードはMAGICの2倍強，と高速です。

このプログラムは横640モード，320モードの両方に対応し，さらに320モードではページ0，1のどちらに描くかを選択可能です。

入力/使用方法

S-OSから入力する場合は，MACINTO-Cなどのマシン語入力ツールから入力してください。また，BASICから入力するには以下の手順に従ってください。

CLEAR　&HE000
MON
ME000

これだけでもよいのですが，プログラムによってはアドレスが適当でない場合がありますので，BASICから使いやすいB000$_H$版も作成しておきましょう。作業にはZING，RINGなどが必要です。具体的な手順は表4にまとめておきました（これはZINGとZEDAを使用した例です）。

S-OSなどから使う場合は，単純に先ほど入力したオブジェクトファイルをロードすればよいのですが，BASICから使う場合はE000$_H$版とB000$_H$版とでロードの仕方が異なります。

・E000$_H$版を使用するとき

CLEAR　&HE000
LOADM“ファイルネーム”

・B000$_H$版を使用するとき

NEWON　&HB500
LOADM　“ファイルネーム”

さて，具体的な使い方に移ることにしましょう。ちなみに文中のE***$_H$をB***$_H$に変えるだけでB000$_H$版に対応しますので，以下ではE000$_H$版に話を絞って説明します。また，始点のX座標をX1，始点のY座標をY1，終点のX座標をX2，終点のY座標をY2と呼ぶことにします。

1）　画面モードの違い

640×200モードを使う場合には画面を80桁に，320×200モードを使う場合には画面を40桁にします（この処理は各ユーザーの用意したプログラムまたはBASICのWIDTHコマンドを使う)。その後，以下のアドレスをコールしてください。

・640×200モード：　CALL　E013$_H$
・320×200モード：　CALL　E010$_H$

2）　ラインのモードを選択

ラインのモードを「ORモード」，「XORモード」のどちらかを選び，以下のアドレスをコールします。

・ORモード　：　CALL　E00A$_H$
・XORモード：　CALL　E00D$_H$

3）　グラフィックプレーンを選択

ラインを描くグラフィックプレーンを選択します。

・Bプレーンページ0：　CALL　E016$_H$
・Bプレーンページ1：　CALL　E019$_H$(320×200モード専用)
・Rプレーンページ0：　CALL　E01C$_H$
・Rプレーンページ1：　CALL　E01F$_H$(320×200モード専用)
・Gプレーンページ0：　CALL　E022$_H$
・Gプレーンページ1：　CALL　E025$_H$(320×200モード専用)

4）　ラインを描く（表1，表2参照）

以上が使い方の手順です。アドレスとその機能の対応を表3に挙げておきました。実際にBASICの「LINE(0,0)−(639,199),XOR,1」相当の線を引くプログラムが図1（レジスタ直接方式）と図2（メモリコマンド方式）です。

表1　E007$_H$番地のCALL方法

レジスタ		
DE	←X1	座標1
B	←Y1	座標1
HL	←X2	座標2
C	←Y2	座標2

サンプルプログラム

●サンプル1

WIDTH80で使用してください。画面中央から時計回りの方向に線を引き，モードを変えて3周して止まります。

・BASICから実行する場合

表2　E028$_H$メモリコマンド，フォーマット

先頭アドレス→		
Y1	(1バイト)	
X1	(2バイト)	
Y2	(1バイト)	
X2	(2バイト)	計6バイト

注）先頭アドレスはE030$_H$に書き込んでおくこと

図1　レジスタ直接方式

```
D000                 1         ORG      0D000H
D000                 2
D000                 3         ;LINE(0,0)-(639,199),XOR,1   ver 2
D000                 4
D000 CD 13 E0        5         CALL     0E013H          ;640x200ﾄﾞｯﾄﾓｰﾄﾞ
D003 CD 0D E0        6         CALL     0E00DH          ;XORﾓｰﾄﾞ
D006 CD 16 E0        7         CALL     0E016H          ;BLUEﾌﾟﾚｰﾝ ﾆ ｶｸ
D009 21 13 D0        8         LD       HL,COM          ;ｺﾏﾝﾄﾞﾚﾂ ﾉ ｱﾄﾞﾚｽ
D00C 22 30 E0        9         LD       (0E030H),HL     ;ｺﾏﾝﾄﾞﾚﾂ ﾉ ｱﾄﾞﾚｽ ｦ ﾜｰｸﾆ ｾｯﾄ
D00F                10
D00F CD 28 E0       11         CALL     0E028H          ;ﾗｲﾝ ｦ ｶｸ
D012                12
D012 C9             13         RET
D013                14
D013                15 COM:
D013 00             16         DB       0               ;Y1
D014 00 00          17         DW       0               ;X1
D016 C7             18         DB       199             ;Y2
D017 7F 02          19         DW       639             ;X2
```

図2　メモリコマンド方式

```
D000                 1         ORG      0D000H
D000                 2
D000                 3         ;LINE(0,0)-(639,199),XOR,1    ver1
D000                 4
D000 CD 13 E0        5         CALL     0E013H          ;640x200ﾄﾞｯﾄﾓｰﾄﾞ
D003 CD 0D E0        6         CALL     0E00DH          ;XORﾓｰﾄﾞ
D006 CD 16 E0        7         CALL     0E016H          ;BLUEﾌﾟﾚｰﾝ ﾆ ｶｸ
D009                 8
D009 11 00 00        9         LD       DE,0            ;X1=0
D00C 21 7F 02       10         LD       HL,639          ;X2=639
D00F 01 C7 00       11         LD       BC,199          ;Y1=  0=C
D012                12                                  ;Y2=199=B
D012 CD 07 E0       13         CALL     0E007H          ;ﾗｲﾝ ｦｶｸ
D015                14
D015 C9             15         RET
```

好きなBASICを立ち上げ，CLEAR &HD000を実行後，E000H版を組み込み，このサンプルプログラムを入力，または入力しLコマンドでE000H版をロード。次にこのサンプルプログラムを入力，または入力しておいたものをLコマンドでロードし，JD000でOK。

●サンプル2

西川善司氏による祝一平氏の顔をグルグル回すプログラムです。実行の仕方は，BASIC CZ-8FB01もしくはCZ-8CB01を立ち上げ，

NEWON &HF600

としたのち，高速ラインルーチンのE000H版をBASICに組み込みます。

初めに"DEMO DATA MAKE.BAS"を入力してRUNしてください。5分くらいすると作業を終了するので，次に，

SAVEM"DATA.OBJ",&HC900,&HFAFF

でセーブしてください。

次に機械語部分を入力します。まず，

CLEAR &HF000

として，

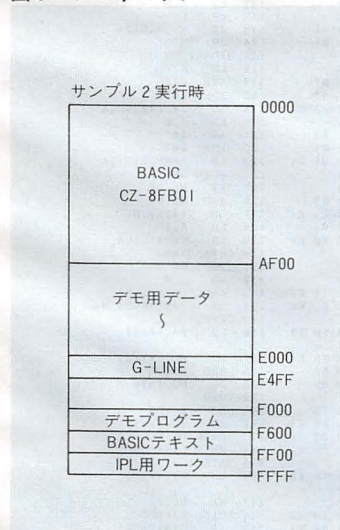
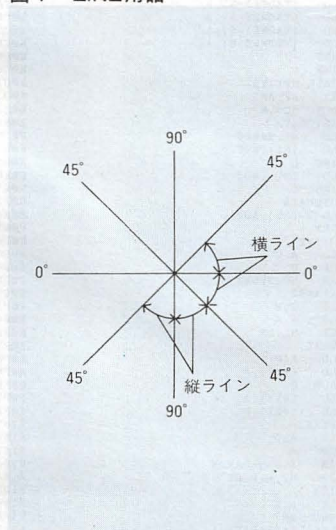
MON

MF000

としてから入力を始めてください。MACINTO-Cなどを使って入力ミスのないことを確認したら，

SAVEM"DEMO.EXE",&HF000,&HF0FF

でセーブしてください。最後に"DEMO RUN.BAS"を入力してRUNで実行です。

また，"DEMO DATA MAKE.BAS"の210～220行の＊Bを取るとまた少し違った動きが楽しめますし，210行のほうをY1＝とし，220行のほうをX1＝とすれば逆回転となります（ほら，高校の数学でやった1次変換ですよ）。

プログラムについて

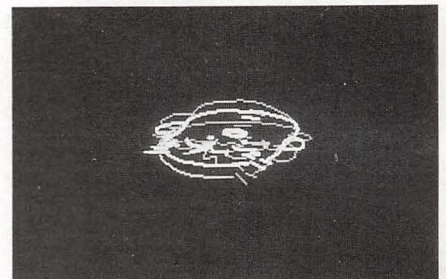
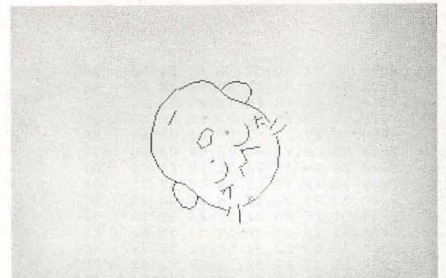
まず，LINEのメインルーチンを8倍に展開しました。それによって各部分が0～7のビットに対応し，それ専用のビットマスクを持ったため，ローテートレジスタやアドレス移動チェックの不要，LINEエンドチェックの時間が1/8になるなどのメリットが得られました。

さらにLINEエンドチェックにA'レジスタを使用するなどして，空いたHLレジスタとDEレジスタを増分カウンタとして使用し，高速化を図っています。

また，横LINE（0°～45°）では，アドレスが変化しない限りIN/OUTしないというアルゴリズムを用いています。これにより，初めて真横の線のみ垂直型のVRAMより速く描けるようになりました。

気になるスピードですが，（0，0）－（639，199）のLINEではMAGICの50本/秒に対して105本/秒と2.1倍，（0,0）－（639,0）ではMAGICの60本/秒に対して157本/秒と2.6倍の速さを示しました。

しかし，このプログラムではクリッピング処理などは行っていませんので，規定外の数値をパラメータとした場合の動作保証はしません。したがって，3Dのワイヤフレームグラフィックなどで使用する際には，クリッピング処理を自分で用意するなどの注意が必要です。

このプログラムのアルゴリズムはまったく機種に依存していないので，簡単に他機種に移植可能です。

速すぎてブレてしまいました

B000H版はMAGICと重なってしまいますが，下位コンパチにしたい人はプログラムの先頭7バイトにおまけ（LD IX,（0C200H）JP DAMY）をつけたので使ってください。

また，このプログラムはサイズを小さくするため，内部書き換えの嵐となっています（Z80の正しい姿？）。

本プログラムはPDSとします。

Profile

◇大野さんは高知県にお住まいの19歳，大学2年生です。マイコン歴7年，1988年5月号の「ON INTERVAL CALL」の作者でもあります。

表3 諸機能

アドレス	NAME	機能
E007	LINE	線を引く（レジスタ直接方式）
E00A	SET「OR」	LINEのモードを「OR」にする
E00D	SET「XOR」	〃 「XOR」にする
E010	SET「320」	横320ドットモードで使用可にする
E013	SET「640」	横640ドットモード 〃
E016	SET「B」	青画面（ページ0）に描く
E019	SET「B1」	青画面ページ1に描く（320モード専用）
E01C	SET「R」	赤画面（ページ0）に描く
E01F	SET「R1」	赤画面ページ1に描く（320モード専用）
E022	SET「G」	緑画面（ページ0）に描く
E025	SET「G1」	緑画面ページ1に描く（320モード専用）
E028	MEMCOM	線を引く（メモリコマンド式）
E030	♯CMDATA	MEMCOMのワークエリア。コマンドの先頭アドレスを入力する

表4 E000H版をB000H～へ移動させる方法（例）

① ZINGでE000H～E4FFHをジェネレート
② ZEDAに入りORGを$B000にする
③ B↵
　C▮$E3▮$B3▮↵
④ B↵
　C▮$E2▮$B2▮↵
⑤ P8↵
　C▮♯E▮$B▮↵
⑥ アセンブルする（ラベルエラーがすべてのラベルで出るが害はない）
⑦ B000H～B4FFHをセーブ
以上（▮は任意の区切り文字を表す）。

図3 メモリマップ　**図4 LINE用語**

リスト1 G-LINE

```
E000 DD 2A 00 C2 C3 F3 E4 C3 : 26
E008 86 E0 C3 32 E0 C3 38 E0 : 16
E010 C3 3E E0 C3 46 E0 C3 AA : 37
E018 E4 C3 B0 E4 C3 B6 E4 C3 : 5B
E020 BC E4 C3 C2 E4 C3 C8 E4 : 78
E028 C3 CE E4 00 00 00 00 00 : 75
E030 00 D0 3E F6 32 EE E4 C9 : D1
E038 3E EE 32 EE E4 C9 AF 32 : DA
E040 38 E1 3E 28 18 0F 3A 21 : 01
E048 E1 E6 F0 32 21 E1 3E 29 : 52
E050 32 38 E1 3E 50 32 38 E2 : 25
E058 32 53 E2 32 6E E2 32 89 : A4
E060 E2 32 A4 E2 32 BF E2 32 : 9F
E068 DA E2 32 F5 E2 32 23 E3 : FD
E070 32 3D E3 32 57 E3 32 71 : 61
E078 E3 32 8B E3 32 A5 E3 32 : 6F
---------------------------------
SUM: 15 50 9F F7 3A 43 1A 5C 8511

E080 BF E3 32 D9 E3 C9 CD 98 : BE
E088 E0 08 ED 79 21 00 00 36 : A5
E090 19 2C 36 D2 2C 36 00 C9 : 78
E098 79 B8 30 03 48 47 EB 79 : 57
E0A0 32 F1 E4 90 32 EB E4 22 : BA
E0A8 EF E4 7A BC 28 1C D2 D0 : EF
E0B0 E0 7B E6 07 32 E7 E4 7D : C2
E0B8 E6 07 32 E5 E4 D5 AF ED : 59
E0C0 52 22 E9 E4 32 ED E4 C3 : 07
E0C8 EC E0 7B BD 30 02 18 E1 : 2F
E0D0 7B E6 07 EE 07 32 E7 E4 : 5A
E0D8 7D E6 07 EE 07 32 E5 E4 : 5A
E0E0 D5 EB ED 52 22 E9 E4 3E : 2C
E0E8 01 32 ED E4 D1 68 CD 15 : 1F
E0F0 E1 22 E3 E4 ED 5B E9 E4 : DF
E0F8 2A EB E4 ED 52 30 0B 42 : B5
---------------------------------
SUM: 2F 1E 0E E3 8A 38 6E 51 3E5B

E100 4B 2A EB E4 CD 3D E1 C3 : F2
E108 58 E1 ED 4B EB E4 EB CD : F8
E110 3D E1 C3 04 E4 CB 3A CB : 99
E118 1B CB 3A CB 1B CB 3B 62 : 6E
E120 16 C0 D5 E5 7D E6 07 C6 : 66
E128 67 29 29 29 44 4D E1 7D : D1
E130 E6 F8 6F 5F 54 29 29 19 : 6B
E138 29 09 D1 19 C9 11 00 00 : F6
E140 3E 10 CB 23 CB 12 ED 6A : 70
E148 ED 42 38 06 1C 3D C2 42 : CA
E150 E1 C9 09 3D C2 42 E1 C9 : 9E
E158 2A E9 E4 CB 3C CB 1D CB : B1
E160 3C CB 1D CB 3D 08 7D 3C : ED
E168 08 3A ED E4 B7 C2 A3 E1 : 10
E170 2A EE E4 26 80 22 25 E2 : CB
E178 26 40 22 40 E2 26 20 22 : 12
---------------------------------
SUM: 51 D8 13 CA D0 92 64 20 2CC1

E180 5B E2 26 10 22 76 E2 26 : 13
E188 08 22 91 E2 26 04 22 AC : 95
E190 E2 26 02 22 C7 E2 26 01 : FC
E198 22 E2 E2 3E 03 32 22 E2 : 5D
E1A0 C3 D3 E1 2A EE E4 26 01 : 9A
E1A8 22 25 E2 26 02 22 40 E2 : 95
E1B0 26 04 22 5B E2 26 08 22 : D9
E1B8 76 E2 26 10 22 91 E2 26 : 49
E1C0 20 22 AC E2 26 40 22 C7 : 1F
E1C8 E2 26 80 22 E2 E2 3E 0B : B7
E1D0 32 22 E2 2A E5 E4 01 1A : 44
E1D8 E2 09 6E 22 8D E0 36 C3 : E1
E1E0 2C 36 FD 2C 7E 32 96 E0 : B1
E1E8 36 E2 2C 2C 22 06 E3 2A : A5
E1F0 E5 E4 01 12 E2 09 6E 26 : 5B
E1F8 E2 22 03 E3 2A E7 E4 0B : EA
---------------------------------
SUM: 27 7B 4F AA 2C 59 FE CA AF64

E200 09 6E E5 ED 4B E3 E4 21 : 7C
E208 00 80 FD 21 22 E2 ED 78 : 07
E210 C9 25 40 5B 76 91 AC C7 : 03
E218 E2 22 27 42 5D 78 93 AE : 83
E220 C9 E6 03 ED 78 F6 80 19 : A6
E228 D2 40 E2 ED 79 78 C6 08 : A0
E230 47 E6 38 C2 3E E2 79 C6 : 86
E238 50 4F 78 CE C0 47 ED 78 : 51
E240 F6 40 19 D2 5B E2 ED 79 : C4
E248 78 C6 08 47 E6 38 C2 59 : C6
E250 E2 79 C6 50 4F 78 CE C0 : C6
E258 47 ED 78 F6 20 19 D2 76 : 23
E260 E2 ED 79 78 C6 08 47 E6 : BB
E268 38 C2 74 E2 79 C6 50 4F : 2E
E270 78 CE C0 47 ED 78 F6 10 : B8
E278 19 D2 91 E2 ED 79 78 C6 : 02
---------------------------------
SUM: 28 4B 7B F7 F8 CF 10 80 293E

E280 08 47 E6 38 C2 8F E2 79 : 19
E288 C6 50 4F 78 CE C0 47 ED : 9F
E290 78 F6 08 19 D2 AC E2 ED : DC
E298 79 78 C6 08 47 E6 38 C2 : E6
E2A0 AA E2 79 C6 50 4F 78 CE : B0
E2A8 C0 47 ED 78 F6 04 19 D2 : 51
E2B0 C7 E2 ED 79 78 C6 08 47 : 9C
E2B8 E6 38 C2 C5 E2 79 C6 50 : 16
E2C0 4F 78 CE C0 47 ED 78 F6 : F7
E2C8 02 19 D2 E2 E2 ED 79 78 : 8F
E2D0 C6 08 47 E6 38 C2 E0 E2 : B7
E2D8 79 C6 50 4F 78 CE C0 47 : 2B
E2E0 ED 78 F6 01 ED 79 19 D2 : AD
E2E8 22 E2 78 C6 08 47 E6 38 : AF
E2F0 C2 22 E2 79 C6 50 4F 78 : 1C
E2F8 CE C0 47 FD E9 08 3D C8 : C8
---------------------------------
SUM: 05 E3 E6 61 C6 F5 BE 2D 7757

E300 08 19 D2 00 00 C3 00 00 : B6
E308 12 2C 46 60 7A 94 AE C8 : 68
E310 00 00 ED 78 F6 80 ED 79 : 41
E318 78 C6 08 47 E6 38 C2 29 : 96
E320 E3 79 C6 50 4F 78 CE C0 : C7
E328 47 19 30 E6 ED 78 F6 40 : 11
E330 ED 79 78 C6 08 47 E6 38 : 11
E338 C2 43 E3 79 C6 50 4F 78 : 3E
E340 CE C0 47 19 30 E6 ED 78 : 69
E348 F6 20 ED 79 78 C6 08 47 : 09
E350 E6 38 C2 5D E3 79 C6 50 : AF
E358 4F 78 CE C0 47 19 30 E6 : CB
E360 ED 78 F6 10 ED 79 78 C6 : 0F
E368 08 47 E6 38 C2 77 E3 79 : 02
E370 C6 50 4F 78 CE C0 47 19 : CB
E378 30 E6 ED 78 F6 08 ED 79 : DF
---------------------------------
SUM: 4F DE 3A 7B A5 8C D0 E0 60B5

E380 78 C6 08 47 E6 38 C2 91 : FE
E388 E3 79 C6 50 4F 78 CE C0 : C7
E390 47 19 30 E6 ED 78 F6 04 : D5
E398 ED 79 78 C6 08 47 E6 38 : 11
E3A0 C2 AB E3 79 C6 50 4F 78 : A6
E3A8 CE C0 47 19 30 E6 ED 78 : 69
E3B0 F6 02 ED 79 78 C6 08 47 : EB
E3B8 E6 38 C2 C5 E3 79 C6 50 : 17
E3C0 4F 78 CE C0 47 19 30 E6 : CB
E3C8 ED 78 F6 01 ED 79 78 C6 : 00
E3D0 08 47 E6 38 C2 DF E3 79 : 6A
E3D8 C6 50 4F 78 CE C0 47 19 : CB
E3E0 30 E6 03 FD E9 ED 79 08 : 6D
E3E8 B9 28 03 08 78 C9 08 3A : 6F
E3F0 E2 E4 B8 28 02 78 C9 21 : 0A
E3F8 00 00 36 ED 2C 36 79 2C : 2A
---------------------------------
SUM: D0 EF 3C 9E CE 79 0B E1 830D

E400 36 78 E1 C9 D5 ED 5B EF : 64
E408 E4 3A F1 E4 6F CD 15 E1 : 25
E410 22 E1 E4 08 7D 08 D1 E1 : 26
E418 3A ED E4 B7 C2 52 E4 2A : E4
E420 EE E4 26 80 22 14 E3 26 : B7
E428 40 22 2E E3 26 20 22 48 : 23
E430 E3 26 10 22 62 E3 26 08 : AE
E438 22 7C E3 26 04 22 96 E3 : 46
E440 26 02 22 B0 E3 26 01 22 : 26
E448 CA E3 3E 03 32 E2 E3 C3 : A8
E450 82 E4 2A EE E4 26 01 22 : AB
E458 14 E3 26 02 22 2E E3 26 : 78
E460 04 22 48 E3 26 08 22 62 : 03
E468 E3 26 10 22 7C E3 26 20 : E0
E470 22 96 E3 26 40 22 B0 E3 : B6
E478 26 80 22 CA E3 3E 0B 32 : F0
---------------------------------
SUM: 5E 32 EE AF 11 F4 B1 F8 DF19

E480 E2 E3 2A E5 E4 01 08 E3 : A4
E488 09 7E C6 04 6F 22 F8 E3 : BD
E490 36 CD 2C 36 E5 2C 36 E3 : 8F
E498 2A E7 E4 09 6E E5 ED 4B : 89
E4A0 E3 E4 21 00 80 FD 21 12 : 98
E4A8 E3 C9 3E 40 32 21 E1 C9 : 27
E4B0 3E 44 32 21 E1 C9 3E 80 : 3D
E4B8 32 21 E1 C9 3E 84 32 21 : 12
E4C0 E1 C9 3E C0 32 21 E1 C9 : A5
E4C8 3E C4 32 21 E1 C9 2A 30 : 59
E4D0 E0 46 23 5E 23 56 23 4E : 91
E4D8 23 7E 23 66 6F C3 86 E0 : C2
E4E0 00 00 00 00 00 00 00 00 : 00
E4E8 00 00 00 00 00 00 F6 00 : F6
E4F0 00 00 00 00 00 00 00 00 : 00
E4F8 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: A3 78 28 F7 1C A2 3F 97 D51B
```

リスト2 G-LINEソースリスト

```
0000              1
0000              2 ;#################
0000              3 ;
0000              4 ;     G-LINE
0000              5 ;
0000              6 ;#################
0000              7
E000              8   ORG     $E000
E000              9
E000 DD 2A 00    10  LD   IX,($C200)
E003 C2
E004 C3 F3 E4    11  JP   DAMY
E007 C3 86 E0    12  JP   LINE
E00A C3 32 E0    13  JP   SETOR
E00D C3 38 E0    14  JP   SETXOR
E010 C3 3E E0    15  JP   SET320
E013 C3 46 E0    16  JP   SET640
E016 C3 AA E4    17  JP   SETB
E019 C3 B0 E4    18  JP   SETB1
E01C C3 B6 E4    19  JP   SETR
E01F C3 BC E4    20  JP   SETR1
E022 C3 C2 E4    21  JP   SETG
E025 C3 C8 E4    22  JP   SETG1
E028 C3 CE E4    23  JP   MEMCOM
E02B 00 00 00    24  DS   5
E02E 00 00
E030             25 #CMDATA
E030 00 D0       26  DW   $D000
E032             27 SETOR
E032 3E F6       28  LD   A,$F6
E034 32 EE E4    29  LD   (#ORXOR),A
E037 C9          30  RET
E038             31 SETXOR
E038 3E EE       32  LD   A,$EE
E03A 32 EE E4    33  LD   (#ORXOR),A
E03D C9          34  RET
E03E             35 SET320
E03E AF          36  XOR  A
E03F 32 38 E1    37  LD   (NOP320),A
E042 3E 28       38  LD   A,$28
E044 18 0F       39  JR   SET50OR28
E046             40 SET640
E046 3A 21 E1    41  LD   A,(BRG+1)
E049 E6 F0       42  AND  $F0
E04B 32 21 E1    43  LD   (BRG+1),A
E04E 3E 29       44  LD   A,$29
E050 32 38 E1    45  LD   (NOP320),A
E053 3E 50       46  LD   A,$50
E055             47 SET50OR28
E055 32 38 E2    48  LD   (YODOWNB7-6),A
E058 32 53 E2    49  LD   (YODOWNB6-6),A
E05B 32 6E E2    50  LD   (YODOWNB5-6),A
E05E 32 89 E2    51  LD   (YODOWNB4-6),A
E061 32 A4 E2    52  LD   (YODOWNB3-6),A
E064 32 BF E2    53  LD   (YODOWNB2-6),A
E067 32 DA E2    54  LD   (YODOWNB1-6),A
E06A 32 F5 E2    55  LD   (YODW28OR50+1),A
E06D 32 23 E3    56  LD   (TADWE7-6),A
E070 32 3D E3    57  LD   (TADWE6-6),A
E073 32 57 E3    58  LD   (TADWE5-6),A
E076 32 71 E3    59  LD   (TADWE4-6),A
E079 32 8B E3    60  LD   (TADWE3-6),A
E07C 32 A5 E3    61  LD   (TADWE2-6),A
E07F 32 BF E3    62  LD   (TADWE1-6),A
E082 32 D9 E3    63  LD   (TADWE0-6),A
E085 C9          64  RET
E086             65 LINE
E086 CD 98 E0    66  CALL CPY1Y2
E089 08          67  EX   AF,AF'
E08A ED 79       68  OUT  (C),A
E08C             69 BASYODATA
E08C 21 00 00    70  LD   HL,$0000
E08F 36 19       71  LD   (HL),$19
E091 2C          72  INC  L
E092 36 D2       73  LD   (HL),$D2
E094 2C          74  INC  L
E095             75 MOTODATA
E095 36 00       76  LD   (HL),$00
E097 C9          77  RET
E098             78 CPY1Y2
E098 79          79  LD   A,C
E099 B8          80  CP   B
E09A 30 03       81  JR   NC,YYSET
E09C             82 EX1AND2
E09C 48          83  LD   C,B
E09D 47          84  LD   B,A
E09E EB          85  EX   HL,DE
E09F             86 YYSET
E09F 79          87  LD   A,C
E0A0 32 F1 E4    88  LD   (#Y2),A
E0A3 90          89  SUB  B
E0A4 32 EB E4    90  LD   (#YY),A
E0A7 22 EF E4    91  LD   (#X2),HL
E0AA             92 CPX1X2
E0AA 7A          93  LD   A,D
E0AB BC          94  CP   H
E0AC 28 1C       95  JR   Z,SAMEDANDH
E0AE D2 D0 E0    96  JP   NC,HIDARI
E0B1             97 MIGI
E0B1 7B          98  LD   A,E
E0B2 E6 07       99  AND  $07
E0B4 32 E7 E4   100  LD   (#STABIT),A
E0B7 7D         101  LD   A,L
E0B8 E6 07      102  AND  $07
E0BA 32 E5 E4   103  LD   (#ENDBIT),A
E0BD D5         104  PUSH DE
E0BE AF         105  XOR  A
E0BF ED 52      106  SBC  HL,DE
E0C1 22 E9 E4   107  LD   (#XX),HL
E0C4 32 ED E4   108  LD   (#SAYU),A
E0C7 C3 EC E0   109  JP   GETADDRESS
E0CA            110 SAMEDANDH
E0CA 7B         111  LD   A,E
E0CB BD         112  CP   L
E0CC 30 02      113  JR   NC,HIDARI
E0CE 18 E1      114  JR   MIGI
E0D0            115 HIDARI
E0D0 7B         116  LD   A,E
E0D1 E6 07      117  AND  $07
E0D3 EE 07      118  XOR  $07
E0D5 32 E7 E4   119  LD   (#STABIT),A
E0D8 7D         120  LD   A,L
E0D9 E6 07      121  AND  $07
E0DB EE 07      122  XOR  $07
E0DD 32 E5 E4   123  LD   (#ENDBIT),A
E0E0 D5         124  PUSH DE
E0E1 EB         125  EX   HL,DE
E0E2 ED 52      126  SBC  HL,DE
E0E4 22 E9 E4   127  LD   (#XX),HL
E0E7 3E 01      128  LD   A,1
E0E9 32 ED E4   129  LD   (#SAYU),A
E0EC            130 GETADDRESS
E0EC D1         131  POP  DE
E0ED 68         132  LD   L,B
E0EE CD 15 E1   133  CALL GETADDHL
E0F1 22 E3 E4   134  LD   (#STAADD),HL
E0F4            135 CPXXYY
E0F4 ED 5B E9   136  LD   DE,(#XX)
E0F7 E4
E0F8 2A EB E4   137  LD   HL,(#YY)
E0FB ED 52      138  SBC  HL,DE
E0FD 30 0B      139  JR   NC,TATE
E0FF            140 YOKO
E0FF 42 4B      141  LD   BC,DE
E101 2A EB E4   142  LD   HL,(#YY)
E104 CD 3D E1   143  CALL DIVHLDE
E107 C3 58 E1   144  JP   LINEYOKO
E10A            145 TATE
E10A ED 4B EB   146  LD   BC,(#YY)
E10D E4
E10E EB         147  EX   HL,DE
E10F CD 3D E1   148  CALL DIVHLDE
E112 C3 04 E4   149  JP   LINETATE
```

▶この前，去年の12月号の「あぶない福袋」にあった，プレゼントの缶詰が届きましたので，さっそく食べてみました。感想は「思ったよりおいしかった」です。しかし。恐ろしいものを食べて胃がびっくりしたのか，少し気分が悪くなってしまった……。

山口　寛憲（19）愛知県

```
E115                150 GETADDHL
E115 CB 3A          151  SRL  D
E117 CB 1B          152  RR   E
E119 CB 3A          153  SRL  D
E11B CB 1B          154  RR   E
E11D CB 3B          155  SRL  E
E11F 62             156  LD   H,D
E120                157 BRG
E120 16 C0          158  LD   D,$C0
E122 D5             159  PUSH DE
E123 E5             160  PUSH HL
E124 7D             161  LD   A,L
E125 E6 07          162  AND  $07
E127 6C             163  LD   L,H
E128 67             164  LD   H,A
E129 29             165  ADD  HL,HL
E12A 29             166  ADD  HL,HL
E12B 29             167  ADD  HL,HL
E12C 44 4D          168  LD   BC,HL
E12E E1             169  POP  HL
E12F 7D             170  LD   A,L
E130 E6 F8          171  AND  $F8
E132 6F             172  LD   L,A
E133 5F             173  LD   E,A
E134 54             174  LD   D,H
E135 29             175  ADD  HL,HL
E136 29             176  ADD  HL,HL
E137 19             177  ADD  HL,DE
E138                178 NOP320
E138 29             179  ADD  HL,HL
E139 09             180  ADD  HL,BC
E13A D1             181  POP  DE
E13B 19             182  ADD  HL,DE
E13C C9             183  RET
E13D                184 DIVHLDE
E13D 11 00 00       185  LD   DE,$0000
E140 3E 10          186  LD   A,$10
E142                187 DIV1
E142 CB 23          188  SLA  E
E144 CB 12          189  RL   D
E146 ED 6A          190  ADC  HL,HL
E148 ED 42          191  SBC  HL,BC
E14A 38 06          192  JR   C,DIV2
E14C 1C             193  INC  E
E14D 3D             194  DEC  A
E14E C2 42 E1       195  JP   NZ,DIV1
E151 C9             196  RET
E152                197 DIV2
E152 09             198  ADD  HL,BC
E153 3D             199  DEC  A
E154 C2 42 E1       200  JP   NZ,DIV1
E157 C9             201  RET
E158                202 LINEYOKO
E158 2A E9 E4       203  LD   HL,(#XX)
E15B CB 3C          204  SRL  H
E15D CB 1D          205  RR   L
E15F CB 3C          206  SRL  H
E161 CB 1D          207  RR   L
E163 CB 3D          208  SRL  L
E165 08             209  EX   AF,AF'
E166 7D             210  LD   A,L
E167 3C             211  INC  A
E168 08             212  EX   AF,AF'
E169 3A ED E4       213  LD   A,(#SAYU)
E16C B7             214  OR   A
E16D C2 A3 E1       215  JP   NZ,YOHIDARISET
E170                216 YOMIGISET
E170 2A EE E4       217  LD   HL,(#ORXOR)
E173 26 80          218  LD   H,$80
E175 22 25 E2       219  LD   (YOKO7),HL
E178 26 40          220  LD   H,$40
E17A 22 40 E2       221  LD   (YOKO6),HL
E17D 26 20          222  LD   H,$20
E17F 22 5B E2       223  LD   (YOKO5),HL
E182 26 10          224  LD   H,$10
E184 22 76 E2       225  LD   (YOKO4),HL
E187 26 08          226  LD   H,$08
E189 22 91 E2       227  LD   (YOKO3),HL
E18C 26 04          228  LD   H,$04
E18E 22 AC E2       229  LD   (YOKO2),HL
E191 26 02          230  LD   H,$02
E193 22 C7 E2       231  LD   (YOKO1),HL
E196 26 01          232  LD   H,$01
E198 22 E2 E2       233  LD   (YOKO0),HL
E19B 3E 03          234  LD   A,$03
E19D 32 22 E2       235  LD   (YOINCADRS),A
E1A0 C3 D3 E1       236  JP   YOKOSET
E1A3                237 YOHIDARISET
E1A3 2A EE E4       238  LD   HL,(#ORXOR)
E1A6 26 01          239  LD   H,$01
E1A8 22 25 E2       240  LD   (YOKO7),HL
E1AB 26 02          241  LD   H,$02
E1AD 22 40 E2       242  LD   (YOKO6),HL
E1B0 26 04          243  LD   H,$04
E1B2 22 5B E2       244  LD   (YOKO5),HL
E1B5 26 08          245  LD   H,$08
E1B7 22 76 E2       246  LD   (YOKO4),HL
E1BA 26 10          247  LD   H,$10
E1BC 22 91 E2       248  LD   (YOKO3),HL
E1BF 26 20          249  LD   H,$20
E1C1 22 AC E2       250  LD   (YOKO2),HL
E1C4 26 40          251  LD   H,$40
E1C6 22 C7 E2       252  LD   (YOKO1),HL
E1C9 26 80          253  LD   H,$80
E1CB 22 E2 E2       254  LD   (YOKO0),HL
E1CE 3E 0B          255  LD   A,$0B
E1D0 32 22 E2       256  LD   (YOINCADRS),A
E1D3                257 YOKOSET
E1D3 2A E5 E4       258  LD   HL,(#ENDBIT)
E1D6 01 1A E2       259  LD   BC,#YOENDDATA
E1D9 09             260  ADD  HL,BC
E1DA 6E             261  LD   L,(HL)
E1DB 22 8D E0       262  LD   (BASYODATA+1),HL
E1DE 36 C3          263  LD   (HL),$C3
E1E0 2C             264  INC  L
E1E1 36 FD          265  LD   (HL),Check
E1E3 2C             266  INC  L
E1E4 7E             267  LD   A,(HL)
E1E5 32 96 E0       268  LD   (MOTODATA+1),A
E1E8 36 E2          269  LD   (HL),Check/256
E1EA 2C             270  INC  L
E1EB 2C             271  INC  L
E1EC 22 06 E3       272  LD   (CheckRET+1),HL
E1EF 2A E5 E4       273  LD   HL,(#ENDBIT)
E1F2 01 12 E2       274  LD   BC,#YOSTADATA+1
E1F5 09             275  ADD  HL,BC
E1F6 6E             276  LD   L,(HL)
E1F7 26 E2          277  LD   H,Check/256
E1F9 22 03 E3       278  LD   (CheckRET-2),HL
E1FC 2A E7 E4       279  LD   HL,(#STABIT)
E1FF 0B             280  DEC  BC
E200 09             281  ADD  HL,BC
E201 6E             282  LD   L,(HL)
E202 E5             283  PUSH HL
E203 ED 4B E3       284  LD   BC,(#STAADD)
E206 E4
E207 21 00 80       285  LD   HL,$8000
```

```
E20A FD 21 22       286  LD   IY,YOINCADRS
E20D E2
E20E ED 78          287  IN   A,(C)
E210 C9             288  RET
E211                289 #YOSTADATA
E211 25             290  DB   YOKO7
E212 40             291  DB   YOKO6
E213 5B             292  DB   YOKO5
E214 76             293  DB   YOKO4
E215 91             294  DB   YOKO3
E216 AC             295  DB   YOKO2
E217 C7             296  DB   YOKO1
E218 E2             297  DB   YOKO0
E219 22             298  DB   YOINCADRS
E21A                299 #YOENDDATA
E21A 27             300  DB   YOKO7+2
E21B 42             301  DB   YOKO6+2
E21C 5D             302  DB   YOKO5+2
E21D 78             303  DB   YOKO4+2
E21E 93             304  DB   YOKO3+2
E21F AE             305  DB   YOKO2+2
E220 C9             306  DB   YOKO1+2
E221 E6             307  DB   YOKO0+4
E222                308 YOINCADRS
E222 03             309  INC  BC
E223 ED 78          310  IN   A,(C)
E225                311 YOKO7
E225 F6 80          312  OR   $80
E227 19             313  ADD  HL,DE
E228 D2 40 E2       314  JP   NC,YOKO6
E22B                315 YODOWN7
E22B ED 79          316  OUT  (C),A
E22D 78             317  LD   A,B
E22E C6 08          318  ADD  A,$08
E230 47             319  LD   B,A
E231 E6 38          320  AND  $38
E233 C2 3E E2       321  JP   NZ,YODOWNB7
E236 79             322  LD   A,C
E237 C6 50          323  ADD  A,$50
E239 4F             324  LD   C,A
E23A 78             325  LD   A,B
E23B CE C0          326  ADC  A,$C0
E23D 47             327  LD   B,A
E23E                328 YODOWNB7
E23E ED 78          329  IN   A,(C)
E240                330 YOKO6
E240 F6 40          331  OR   $40
E242 19             332  ADD  HL,DE
E243 D2 5B E2       333  JP   NC,YOKO5
E246                334 YODOWN6
E246 ED 79          335  OUT  (C),A
E248 78             336  LD   A,B
E249 C6 08          337  ADD  A,$08
E24B 47             338  LD   B,A
E24C E6 38          339  AND  $38
E24E C2 59 E2       340  JP   NZ,YODOWNB6
E251 79             341  LD   A,C
E252 C6 50          342  ADD  A,$50
E254 4F             343  LD   C,A
E255 78             344  LD   A,B
E256 CE C0          345  ADC  A,$C0
E258 47             346  LD   B,A
E259                347 YODOWNB6
E259 ED 78          348  IN   A,(C)
E25B                349 YOKO5
E25B F6 20          350  OR   $20
E25D 19             351  ADD  HL,DE
E25E D2 76 E2       352  JP   NC,YOKO4
E261                353 YODOWN5
E261 ED 79          354  OUT  (C),A
E263 78             355  LD   A,B
E264 C6 08          356  ADD  A,$08
E266 47             357  LD   B,A
E267 E6 38          358  AND  $38
E269 C2 74 E2       359  JP   NZ,YODOWNB5
E26C 79             360  LD   A,C
E26D C6 50          361  ADD  A,$50
E26F 4F             362  LD   C,A
E270 78             363  LD   A,B
E271 CE C0          364  ADC  A,$C0
E273 47             365  LD   B,A
E274                366 YODOWNB5
E274 ED 78          367  IN   A,(C)
E276                368 YOKO4
E276 F6 10          369  OR   $10
E278 19             370  ADD  HL,DE
E279 D2 91 E2       371  JP   NC,YOKO3
E27C                372 YODOWN4
E27C ED 79          373  OUT  (C),A
E27E 78             374  LD   A,B
E27F C6 08          375  ADD  A,$08
E281 47             376  LD   B,A
E282 E6 38          377  AND  $38
E284 C2 8F E2       378  JP   NZ,YODOWNB4
E287 79             379  LD   A,C
E288 C6 50          380  ADD  A,$50
E28A 4F             381  LD   C,A
E28B 78             382  LD   A,B
E28C CE C0          383  ADC  A,$C0
E28E 47             384  LD   B,A
E28F                385 YODOWNB4
E28F ED 78          386  IN   A,(C)
E291                387 YOKO3
E291 F6 08          388  OR   $08
E293 19             389  ADD  HL,DE
E294 D2 AC E2       390  JP   NC,YOKO2
E297                391 YODOWN3
E297 ED 79          392  OUT  (C),A
E299 78             393  LD   A,B
E29A C6 08          394  ADD  A,$08
E29C 47             395  LD   B,A
E29D E6 38          396  AND  $38
E29F C2 AA E2       397  JP   NZ,YODOWNB3
E2A2 79             398  LD   A,C
E2A3 C6 50          399  ADD  A,$50
E2A5 4F             400  LD   C,A
E2A6 78             401  LD   A,B
E2A7 CE C0          402  ADC  A,$C0
E2A9 47             403  LD   B,A
E2AA                404 YODOWNB3
E2AA ED 78          405  IN   A,(C)
E2AC                406 YOKO2
E2AC F6 04          407  OR   $04
E2AE 19             408  ADD  HL,DE
E2AF D2 C7 E2       409  JP   NC,YOKO1
E2B2                410 YODOWN2
E2B2 ED 79          411  OUT  (C),A
E2B4 78             412  LD   A,B
E2B5 C6 08          413  ADD  A,$08
E2B7 47             414  LD   B,A
E2B8 E6 38          415  AND  $38
E2BA C2 C5 E2       416  JP   NZ,YODOWNB2
E2BD 79             417  LD   A,C
E2BE C6 50          418  ADD  A,$50
E2C0 4F             419  LD   C,A
E2C1 78             420  LD   A,B
E2C2 CE C0          421  ADC  A,$C0
```

```
E2C4 47             422  LD   B,A
E2C5                423 YODOWNB2
E2C5 ED 78          424  IN   A,(C)
E2C7                425 YOKO1
E2C7 F6 02          426  OR   $02
E2C9 19             427  ADD  HL,DE
E2CA D2 E2 E2       428  JP   NC,YOKO0
E2CD                429 YODOWN1
E2CD ED 79          430  OUT  (C),A
E2CF 78             431  LD   A,B
E2D0 C6 08          432  ADD  A,$08
E2D2 47             433  LD   B,A
E2D3 E6 38          434  AND  $38
E2D5 C2 E0 E2       435  JP   NZ,YODOWNB1
E2D8 79             436  LD   A,C
E2D9 C6 50          437  ADD  A,$50
E2DB 4F             438  LD   C,A
E2DC 78             439  LD   A,B
E2DD CE C0          440  ADC  A,$C0
E2DF 47             441  LD   B,A
E2E0                442 YODOWNB1
E2E0 ED 78          443  IN   A,(C)
E2E2                444 YOKO0
E2E2 F6 01          445  OR   $01
E2E4 ED 79          446  OUT  (C),A
E2E6 19             447  ADD  HL,DE
E2E7 D2 22 E2       448  JP   NC,YOINCADRS
E2EA                449 YODOWN0
E2EA 78             450  LD   A,B
E2EB C6 08          451  ADD  A,$08
E2ED 47             452  LD   B,A
E2EE E6 38          453  AND  $38
E2F0 C2 22 E2       454  JP   NZ,YOINCADRS
E2F3 79             455  LD   A,C
E2F4                456 YODW28OR50
E2F4 C6 50          457  ADD  A,$50
E2F6 4F             458  LD   C,A
E2F7 78             459  LD   A,B
E2F8 CE C0          460  ADC  A,$C0
E2FA 47             461  LD   B,A
E2FB FD E9          462  JP   (IY)
E2FD                463 Check
E2FD 08             464  EX   AF,AF'
E2FE 3D             465  DEC  A
E2FF C8             466  RET  Z
E300 08             467  EX   AF,AF'
E301 19             468  ADD  HL,DE
E302 D2 00 00       469  JP   NC,$0000
E305                470 CheckRET
E305 C3 00 00       471  JP   $0000
E308                472 #TATEDATA
E308 12             473  DB   TATE7
E309 2C             474  DB   TATE6
E30A 46             475  DB   TATE5
E30B 60             476  DB   TATE4
E30C 7A             477  DB   TATE3
E30D 94             478  DB   TATE2
E30E AE             479  DB   TATE1
E30F C8             480  DB   TATE0
E310 00 00          481  DW   0
E312                482 TATE7
E312 ED 78          483  IN   A,(C)
E314 F6 80          484  OR   $80
E316 ED 79          485  OUT  (C),A
E318 78             486  LD   A,B
E319 C6 08          487  ADD  A,$08
E31B 47             488  LD   B,A
E31C E6 38          489  AND  $38
E31E C2 29 E3       490  JP   NZ,TADWE7
E321 79             491  LD   A,C
E322 C6 50          492  ADD  A,$50
E324 4F             493  LD   C,A
E325 78             494  LD   A,B
E326 CE C0          495  ADC  A,$C0
E328 47             496  LD   B,A
E329                497 TADWE7
E329 19             498  ADD  HL,DE
E32A 30 E6          499  JR   NC,TATE7
E32C                500 TATE6
E32C ED 78          501  IN   A,(C)
E32E F6 40          502  OR   $40
E330 ED 79          503  OUT  (C),A
E332 78             504  LD   A,B
E333 C6 08          505  ADD  A,$08
E335 47             506  LD   B,A
E336 E6 38          507  AND  $38
E338 C2 43 E3       508  JP   NZ,TADWE6
E33B 79             509  LD   A,C
E33C C6 50          510  ADD  A,$50
E33E 4F             511  LD   C,A
E33F 78             512  LD   A,B
E340 CE C0          513  ADC  A,$C0
E342 47             514  LD   B,A
E343                515 TADWE6
E343 19             516  ADD  HL,DE
E344 30 E6          517  JR   NC,TATE6
E346                518 TATE5
E346 ED 78          519  IN   A,(C)
E348 F6 20          520  OR   $20
E34A ED 79          521  OUT  (C),A
E34C 78             522  LD   A,B
E34D C6 08          523  ADD  A,$08
E34F 47             524  LD   B,A
E350 E6 38          525  AND  $38
E352 C2 5D E3       526  JP   NZ,TADWE5
E355 79             527  LD   A,C
E356 C6 50          528  ADD  A,$50
E358 4F             529  LD   C,A
E359 78             530  LD   A,B
E35A CE C0          531  ADC  A,$C0
E35C 47             532  LD   B,A
E35D                533 TADWE5
E35D 19             534  ADD  HL,DE
E35E 30 E6          535  JR   NC,TATE5
E360                536 TATE4
E360 ED 78          537  IN   A,(C)
E362 F6 10          538  OR   $10
E364 ED 79          539  OUT  (C),A
E366 78             540  LD   A,B
E367 C6 08          541  ADD  A,$08
E369 47             542  LD   B,A
E36A E6 38          543  AND  $38
E36C C2 77 E3       544  JP   NZ,TADWE4
E36F 79             545  LD   A,C
E370 C6 50          546  ADD  A,$50
E372 4F             547  LD   C,A
E373 78             548  LD   A,B
E374 CE C0          549  ADC  A,$C0
E376 47             550  LD   B,A
E377                551 TADWE4
E377 19             552  ADD  HL,DE
E378 30 E6          553  JR   NC,TATE4
E37A                554 TATE3
E37A ED 78          555  IN   A,(C)
E37C F6 08          556  OR   $08
E37E ED 79          557  OUT  (C),A
E380 78             558  LD   A,B
```

▶6月21日，CCD-TR55 (790gのHandycam) を買ってしまいました。現在，喜んでポコリン (猫) とパカリン (犬) を撮りまくっています。画質は「こんなもんかな」って，感じだけどかわいいから許しちゃう！ さあ次はX68000だ(こればっか)。ちなみに「なまはげ」とは私の兄です。
大脇 寿法 (21) 愛知県

```
E381 C6 08        559  ADD  A,$08
E383 47           560  LD   B,A
E384 E6 38        561  AND  $38
E386 C2 91 E3     562  JP   NZ,TADWE3
E389 79           563  LD   A,C
E38A C6 50        564  ADD  A,$50
E38C 4F           565  LD   C,A
E38D 78           566  LD   A,B
E38E CE C0        567  ADC  A,$C0
E390 47           568  LD   B,A
E391              569 TADWE3
E391 19           570  ADD  HL,DE
E392 30 E6        571  JR   NC,TATE3
E394              572 TATE2
E394 ED 78        573  IN   A,(C)
E396 F6 04        574  OR   $04
E398 ED 79        575  OUT  (C),A
E39A 78           576  LD   A,B
E39B C6 08        577  ADD  A,$08
E39D 47           578  LD   B,A
E39E E6 38        579  AND  $38
E3A0 C2 AB E3     580  JP   NZ,TADWE2
E3A3 79           581  LD   A,C
E3A4 C6 50        582  ADD  A,$50
E3A6 4F           583  LD   C,A
E3A7 78           584  LD   A,B
E3A8 CE C0        585  ADC  A,$C0
E3AA 47           586  LD   B,A
E3AB              587 TADWE2
E3AB 19           588  ADD  HL,DE
E3AC 30 E6        589  JR   NC,TATE2
E3AE              590 TATE1
E3AE ED 78        591  IN   A,(C)
E3B0 F6 02        592  OR   $02
E3B2 ED 79        593  OUT  (C),A
E3B4 78           594  LD   A,B
E3B5 C6 08        595  ADD  A,$08
E3B7 47           596  LD   B,A
E3B8 E6 38        597  AND  $38
E3BA C2 C5 E3     598  JP   NZ,TADWE1
E3BD 79           599  LD   A,C
E3BE C6 50        600  ADD  A,$50
E3C0 4F           601  LD   C,A
E3C1 78           602  LD   A,B
E3C2 CE C0        603  ADC  A,$C0
E3C4 47           604  LD   B,A
E3C5              605 TADWE1
E3C5 19           606  ADD  HL,DE
E3C6 30 E6        607  JR   NC,TATE1
E3C8              608 TATE0
E3C8 ED 78        609  IN   A,(C)
E3CA F6 01        610  OR   $01
E3CC ED 79        611  OUT  (C),A
E3CE 78           612  LD   A,B
E3CF C6 08        613  ADD  A,$08
E3D1 47           614  LD   B,A
E3D2 E6 38        615  AND  $38
E3D4 C2 DF E3     616  JP   NZ,TADWE0
E3D7 79           617  LD   A,C
E3D8 C6 50        618  ADD  A,$50
E3DA 4F           619  LD   C,A
E3DB 78           620  LD   A,B
E3DC CE C0        621  ADC  A,$C0
E3DE 47           622  LD   B,A
E3DF              623 TADWE0
E3DF 19           624  ADD  HL,DE
E3E0 30 E6        625  JR   NC,TATE0
E3E2              626 TAINCADRS
E3E2 03           627  INC  BC
E3E3 FD E9        628  JP   (IY)
E3E5              629 Check2
E3E5 ED 79        630  OUT  (C),A
E3E7 08           631  EX   AF,AF'
E3E8 B9           632  CP   C
E3E9 28 03        633  JR   Z,CheckR
E3EB 08           634  EX   AF,AF'
E3EC 78           635  LD   A,B
E3ED C9           636  RET
E3EE              637 CheckR
E3EE 08           638  EX   AF,AF'
E3EF 3A E2 E4     639  LD   A,(#ENDB)
E3F2 B8           640  CP   B
E3F3 28 02        641  JR   Z,TATERET
E3F5 78           642  LD   A,B
E3F6 C9           643  RET
E3F7              644 TATERET
E3F7 21 00 00     645  LD   HL,$0000
E3FA 36 ED        646  LD   (HL),$ED
E3FC 2C           647  INC  L
E3FD 36 79        648  LD   (HL),$79
E3FF 2C           649  INC  L
E400 36 78        650  LD   (HL),$78
E402 E1           651  POP  HL
E403 C9           652  RET
E404              653 LINETATE
E404 D5           654  PUSH DE
E405 ED 5B EF     655  LD   DE,(#X2)
E408 E4
E409 3A F1 E4     656  LD   A,(#Y2)
E40C 6F           657  LD   L,A
E40D CD 15 E1     658  CALL GETADDHL
E410 22 E1 E4     659  LD   (#ENDB-1),HL
E413 08           660  EX   AF,AF'
E414 7D           661  LD   A,L
E415 08           662  EX   AF,AF'
E416 D1           663  POP  DE
E417 E1           664  POP  HL
E418 3A ED E4     665  LD   A,(#SAYU)
E41B B7           666  OR   A
E41C C2 52 E4     667  JP   NZ,TAHIDARISET
E41F              668 TAMIGISET
E41F 2A EE E4     669  LD   HL,(#ORXOR)
E422 26 80        670  LD   H,$80
E424 22 14 E3     671  LD   (TATE7+2),HL
E427 26 40        672  LD   H,$40
E429 22 2E E3     673  LD   (TATE6+2),HL
E42C 26 20        674  LD   H,$20
E42E 22 48 E3     675  LD   (TATE5+2),HL
E431 26 10        676  LD   H,$10
E433 22 62 E3     677  LD   (TATE4+2),HL
E436 26 08        678  LD   H,$08
E438 22 7C E3     679  LD   (TATE3+2),HL
E43B 26 04        680  LD   H,$04
E43D 22 96 E3     681  LD   (TATE2+2),HL
E440 26 02        682  LD   H,$02
E442 22 B0 E3     683  LD   (TATE1+2),HL
E445 26 01        684  LD   H,$01
E447 22 CA E3     685  LD   (TATE0+2),HL
E44A 3E 03        686  LD   A,$03
E44C 32 E2 E3     687  LD   (TAINCADRS),A
E44F C3 82 E4     688  JP   TATESET
E452              689 TAHIDARISET
E452 2A EE E4     690  LD   HL,(#ORXOR)
E455 26 01        691  LD   H,$01
E457 22 14 E3     692  LD   (TATE7+2),HL
E45A 26 02        693  LD   H,$02
E45C 22 2E E3     694  LD   (TATE6+2),HL
E45F 26 04        695  LD   H,$04
E461 22 48 E3     696  LD   (TATE5+2),HL
E464 26 08        697  LD   H,$08
E466 22 62 E3     698  LD   (TATE4+2),HL
E469 26 10        699  LD   H,$10
E46B 22 7C E3     700  LD   (TATE3+2),HL
E46E 26 20        701  LD   H,$20
E470 22 96 E3     702  LD   (TATE2+2),HL
E473 26 40        703  LD   H,$40
E475 22 B0 E3     704  LD   (TATE1+2),HL
E478 26 80        705  LD   H,$80
E47A 22 CA E3     706  LD   (TATE0+2),HL
E47D 3E 0B        707  LD   A,$0B
E47F 32 E2 E3     708  LD   (TAINCADRS),A
E482              709 TATESET
E482 2A E5 E4     710  LD   HL,(#ENDBIT)
E485 01 08 E3     711  LD   BC,#TATEDATA
E488 09           712  ADD  HL,BC
E489 7E           713  LD   A,(HL)
E48A C6 04        714  ADD  A,4
E48C 6F           715  LD   L,A
E48D 22 F8 E3     716  LD   (TATERET+1),HL
E490 36 CD        717  LD   (HL),$CD
E492 2C           718  INC  L
E493 36 E5        719  LD   (HL),Check2
E495 2C           720  INC  L
E496 36 E3        721  LD   (HL),Check2/256
E498 2A E7 E4     722  LD   HL,(#STABIT)
E49B 09           723  ADD  HL,BC
E49C 6E           724  LD   L,(HL)
E49D E5           725  PUSH HL
E49E ED 4B E3     726  LD   BC,(#STAADD)
E4A1 E4
E4A2 21 00 80     727  LD   HL,$8000
E4A5 FD 21 12     728  LD   IY,TATE7
E4A8 E3
E4A9 C9           729  RET
E4AA              730 SETB
E4AA 3E 40        731  LD   A,$40
E4AC 32 21 E1     732  LD   (BRG+1),A
E4AF C9           733  RET
E4B0              734 SETB1
E4B0 3E 44        735  LD   A,$44
E4B2 32 21 E1     736  LD   (BRG+1),A
E4B5 C9           737  RET
E4B6              738 SETR
E4B6 3E 80        739  LD   A,$80
E4B8 32 21 E1     740  LD   (BRG+1),A
E4BB C9           741  RET
E4BC              742 SETR1
E4BC 3E 84        743  LD   A,$84
E4BE 32 21 E1     744  LD   (BRG+1),A
E4C1 C9           745  RET
E4C2              746 SETG
E4C2 3E C0        747  LD   A,$C0
E4C4 32 21 E1     748  LD   (BRG+1),A
E4C7 C9           749  RET
E4C8              750 SETG1
E4C8 3E C4        751  LD   A,$C4
E4CA 32 21 E1     752  LD   (BRG+1),A
E4CD C9           753  RET
E4CE              754 MEMCOM
E4CE 2A 30 E0     755  LD   HL,(#CMDATA)
E4D1 46           756  LD   B,(HL)
E4D2 23           757  INC  HL
E4D3 5E           758  LD   E,(HL)
E4D4 23           759  INC  HL
E4D5 56           760  LD   D,(HL)
E4D6 23           761  INC  HL
E4D7 4E           762  LD   C,(HL)
E4D8 23           763  INC  HL
E4D9 7E           764  LD   A,(HL)
E4DA 23           765  INC  HL
E4DB 66           766  LD   H,(HL)
E4DC 6F           767  LD   L,A
E4DD C3 86 E0     768  JP   LINE
E4E0              769 #ENDADD
E4E0 00           770  DB   $00
E4E1 00           771  DB   $00
E4E2              772 #ENDB
E4E2 00           773  DB 0
E4E3              774 #STAADD
E4E3 00 00        775  DW 0
E4E5              776 #ENDBIT
E4E5 00 00        777  DW 0
E4E7              778 #STABIT
E4E7 00 00        779  DW 0
E4E9              780 #XX
E4E9 00 00        781  DW 0
E4EB              782 #YY
E4EB 00 00        783  DW 0
E4ED              784 #SAYU
E4ED 00           785  DB 0
E4EE              786 #ORXOR
E4EE F6           787  DB $F6
E4EF              788 #X2
E4EF 00 00        789  DW 0
E4F1              790 #Y2
E4F1 00 00        791  DW 0
E4F3              792 DAMY
E4F3              793 ;
```

リスト3 サンプル1

```
D000 CD 13 E0 CD 16 E0 CD 0A : 5A
D008 E0 CD 19 D0 CD 0D E0 CD : 1D
D010 19 D0 CD 1C E0 CD 19 D0 : 68
D018 C9 06 64 11 7F 02 0E 00 : D3
D020 21 81 FD 19 D5 E5 C5 11 : 48
D028 40 01 CD 07 E0 C1 E1 D1 : 68
D030 23 B7 ED 52 20 ED 11 40 : 77
D038 01 21 7F 02 D5 E5 C5 CD : EF
D040 07 E0 C1 E1 D1 0C 3E C7 : 6B
D048 B9 20 F1 11 00 00 D5 E5 : 95
D050 C5 11 40 01 CD 07 E0 C1 : 8C
D058 E1 D1 2B B7 ED 52 20 EE : E1
D060 21 00 00 11 40 01 D5 E5 : 2D
D068 C5 CD 07 E0 C1 E1 D1 0D : F9
D070 20 F4 C9 00 00 00 00 00 : DD
D078 00 00 00 00 00 00 00 00 : 00
-----------------------------------
SUM: 80 B3 4D D9 78 7B 09 E3 E0F8
```

リスト4 サンプル1 ソースリスト

```
0000              1
D000              2   ORG  $D000
D000              3
D000 CD 13 E0     4  CALL $E013
D003 CD 16 E0     5  CALL $E016
D006 CD 0A E0     6  CALL $E00A
D009 CD 19 D0     7  CALL LINE1680
D00C CD 0D E0     8  CALL $E00D
D00F CD 19 D0     9  CALL LINE1680
D012 CD 1C E0    10  CALL $E01C
D015 CD 19 D0    11  CALL LINE1680
D018 C9          12  RET
D019             13 LINE1680
D019 06 64       14  LD   B,100
D01B 11 7F 02    15  LD   DE,639
D01E 0E 00       16  LD   C,0
D020 21 81 FD    17  LD   HL,-639
D023             18 LOOP
D023 19          19  ADD  HL,DE
D024 D5          20  PUSH DE
D025 E5          21  PUSH HL
D026 C5          22  PUSH BC
D027 11 40 01    23  LD   DE,320
D02A CD 07 E0    24  CALL $E007
D02D C1          25  POP  BC
D02E E1          26  POP  HL
D02F D1          27  POP  DE
D030 23          28  INC  HL
D031 B7 ED 52    29  SUB  HL,DE
D034 20 ED       30  JR   NZ,LOOP
D036 11 40 01    31  LD   DE,320
D039 21 7F 02    32  LD   HL,639
D03C             33 LOOP2
D03C D5          34  PUSH DE
D03D E5          35  PUSH HL
D03E C5          36  PUSH BC
D03F CD 07 E0    37  CALL $E007
D042 C1          38  POP  BC
D043 E1          39  POP  HL
D044 D1          40  POP  DE
D045 0C          41  INC  C
D046 3E C7       42  LD   A,199
D048 B9          43  CP   C
D049 20 F1       44  JR   NZ,LOOP2
D04B 11 00 00    45  LD   DE,0
D04E             46 LOOP3
D04E D5          47  PUSH DE
D04F E5          48  PUSH HL
D050 C5          49  PUSH BC
D051 11 40 01    50  LD   DE,320
D054 CD 07 E0    51  CALL $E007
D057 C1          52  POP  BC
D058 E1          53  POP  HL
D059 D1          54  POP  DE
D05A 2B          55  DEC  HL
D05B B7 ED 52    56  SUB  HL,DE
D05E 20 EE       57  JR   NZ,LOOP3
D060 21 00 00    58  LD   HL,0
D063 11 40 01    59  LD   DE,320
D066             60 LOOP4
D066 D5          61  PUSH DE
D067 E5          62  PUSH HL
D068 C5          63  PUSH BC
D069 CD 07 E0    64  CALL $E007
D06C C1          65  POP  BC
D06D E1          66  POP  HL
D06E D1          67  POP  DE
D06F 0D          68  DEC  C
D070 20 F4       69  JR   NZ,LOOP4
D072 C9          70  RET
```

リスト5 サンプル2

```
F000 3E 55 32 B0 F0 CD 0A E0 : 1C
F008 CD 10 E0 CD 16 E0 CD 7A : C7
F010 F0 21 00 AE CD 3E F0 E5 : 9F
F018 21 B0 F0 CB 06 38 0E CD : A5
F020 6E F0 CD 19 E0 01 A9 44 : 12
F028 CD 87 F0 18 0C CD 7A F0 : 9F
F030 CD 16 E0 01 A9 40 CD 87 : 01
F038 F0 E1 23 23 18 D6 22 30 : 57
F040 E0 CD 28 E0 2A 30 E0 11 : 00
F048 07 00 19 7E 2B FE FF C2 : 88
F050 68 F0 7E FE FF C8 FE FE : 97
F058 C2 5F F0 23 23 18 DF FE : 4C
F060 FD C2 68 F0 21 FE AD C9 : AC
F068 2B 2B 2B C3 3E F0 01 00 : 73
F070 18 3E 0C ED 79 0C AF ED : 70
F078 79 C9 01 00 18 3E 0C ED : 92
-----------------------------------
SUM: DE B4 11 6A ED 4D 0C 69 65A3

F080 79 0C 3E 04 ED 79 C9 1E : 14
F088 11 60 69 16 12 AF ED 79 : 17
F090 03 15 C2 8E F0 44 4D 78 : 61
F098 C6 08 07 DA A3 F0 0F 47 : 98
F0A0 C3 89 F0 01 D8 37 B7 ED : F0
F0A8 42 44 4D 1D C2 8B F0 C9 : F6
F0B0 00 00 00 00 00 00 00 00 : 00
F0B8 00 00 00 00 00 00 00 00 : 00
F0C0 00 00 00 00 00 00 00 00 : 00
F0C8 00 00 00 00 00 00 00 00 : 00
F0D0 00 00 00 00 00 00 00 00 : 00
F0D8 00 00 00 00 00 00 00 00 : 00
F0E0 00 00 00 00 00 00 00 00 : 00
F0E8 00 00 00 00 00 00 00 00 : 00
F0F0 00 00 00 00 00 00 00 00 : 00
F0F8 00 00 00 00 00 00 00 00 : 00
-----------------------------------
SUM: 58 56 AD A0 2C 1E B9 0C 1328
```

▶アルシオーネに人気が集中しているようですが，スタリオンとかのオーナーはいらっしゃらないのでしょうか？
斎藤 栄一郎 (23) 埼玉県

リスト6 サンプル2 ソースリスト

```
0000                 1 ;------------------------------
0000                 2 ;            祝さんの
0000                 3 ;          顔をグルグル
0000                 4 ;            まわす!?
0000                 5 ;   PROGRAMMED BY Z.NISHIKAWA
0000                 6 ;------------------------------
0000                 7
F000                 8         ORG     0F000H
F000                 9
F000                10 LINE:           EQU     0E007H
F000                11 MEMCOM:         EQU     0E028H
F000                12 #CMDATA:        EQU     0E030H
F000                13 SETOR:          EQU     0E00AH
F000                14 SETXOR:         EQU     0E00DH
F000                15 SET640:         EQU     0E013H
F000                16 SET320:         EQU     0E010H
F000                17 SETB0:          EQU     0E016H
F000                18 SETB1:          EQU     0E019H
F000                19 DATA:           EQU     0AE00H
F000                20 CLS_ADR:        EQU     4*40+9
F000                21 X_WIDTH:        EQU     18
F000                22 Y_WIDTH:        EQU     17
F000 3E 55          23         LD      A,55H
F002 32 B0 F0       24         LD      (PG_FLG),A
F005 CD 0A E0       25         CALL    SETOR           ;ORﾓｰﾄﾞ
F008 CD 10 E0       26         CALL    SET320          ;320x200ﾓｰﾄﾞ
F00B CD 16 E0       27         CALL    SETB0           ;SCREEN 1のBLUEに描くﾓｰﾄﾞ
F00E CD 7A F0       28         CALL    DSP_PG1         ;SCREEN 1を表示
F011 21 00 AE       29         LD      HL,DATA
F014                30
F014                31 MAIN_LOOP:
F014 CD 3E F0       32         CALL    DRAW            ;描くルーチンへ分岐
F017 E5             33         PUSH    HL
F018                34
F018 21 B0 F0       35         LD      HL,PG_FLG
F01B CB 06          36         RLC     (HL)
F01D 38 0E          37         JR      C,CHANGE_PG     ;SCREEN 0に描くかSCREEN 1に描
くかを決定
F01F                38
F01F CD 6E F0       39         CALL    DSP_PG0         ;SCREEN 0を表示
F022 CD 19 E0       40         CALL    SETB1           ;SCREEN 1のBLUEに描くﾓｰﾄﾞ
F025 01 A9 44       41         LD      BC,CLS_ADR+4400H
F028 CD 87 F0       42         CALL    CLS0            ;SCREEN 1を消去
F02B 18 0C          43         JR      INC_PTR
F02D                44
F02D                45 CHANGE_PG:
F02D CD 7A F0       46         CALL    DSP_PG1         ;SCREEN 1を表示
F030 CD 16 E0       47         CALL    SETB0           ;SCREEN 0のBLUEに描くﾓｰﾄﾞ
F033 01 A9 40       48         LD      BC,CLS_ADR+4000H
F036 CD 87 F0       49         CALL    CLS0            ;SCREEN 0を消去
F039                50 INC_PTR:
F039 E1             51         POP     HL
F03A 23             52         INC     HL
F03B 23             53         INC     HL
F03C 18 D6          54         JR      MAIN_LOOP
F03E                55
F03E                56 DRAW:
F03E                57 LOOP01:
F03E 22 30 E0       58         LD      (#CMDATA),HL    ;座表データのあるアドレスをセ
ット
F041 CD 28 E0       59         CALL    MEMCOM          ;線を引く
F044 2A 30 E0       60         LD      HL,(#CMDATA)
F047 11 07 00       61         LD      DE,7
F04A 19             62         ADD     HL,DE
F04B 7E             63         LD      A,(HL)
F04C 2B             64         DEC     HL
F04D FE FF          65         CP      0FFH
F04F C2 68 F0       66         JP      NZ,NOT_FF@@
F052 7E             67         LD      A,(HL)
F053 FE FF          68         CP      0FFH            ;ﾃﾞｰﾀが65535なら絵1枚描き終っ
た
F055 C8             69         RET     Z               ;EXIT   HL+2 = NEXT COM.
F056 FE FE          70         CP      0FEH            ;ﾃﾞｰﾀが65534なら次の線を引く
F058 C2 5F F0       71         JP      NZ,PIC_END?
F05B 23             72         INC     HL
F05C 23             73         INC     HL
F05D 18 DF          74         JR      LOOP01
F05F                75 PIC_END?:
F05F FE FD          76         CP      0FDH            ;ﾃﾞｰﾀが65533なら全ての絵を描
き終った
F061 C2 68 F0       77         JP      NZ,NOT_FF@@
F064 21 FE AD       78         LD      HL,DATA-2
F067 C9             79         RET
F068                80
F068                81 NOT_FF@@:
F068 2B             82         DEC     HL
F069 2B             83         DEC     HL
F06A 2B             84         DEC     HL
F06B C3 3E F0       85         JP      LOOP01
F06E                86
F06E                87 DSP_PG0:
F06E 01 00 18       88         LD      BC,1800H        ;BC=CRTC I/O ADDR.
F071 3E 0C          89         LD      A,0CH           ;CRTCのﾚｼﾞｽﾀの12番を選択
F073 ED 79          90         OUT     (C),A
F075 0C             91         INC     C
F076 AF             92         XOR     A               ;R12に0をｾｯﾄ.....PAGE 1を表示
F077 ED 79          93         OUT     (C),A
F079 C9             94         RET
F07A                95
F07A                96 DSP_PG1:
F07A 01 00 18       97         LD      BC,1800H        ;BC=CRTC I/O ADDR.
F07D 3E 0C          98         LD      A,0CH           ;CRTCのﾚｼﾞｽﾀの12番を選択
F07F ED 79          99         OUT     (C),A
F081 0C            100         INC     C
F082 3E 04         101         LD      A,4             ;R12に4をｾｯﾄ.....PAGE 1を表示
F084 ED 79         102         OUT     (C),A
F086 C9            103         RET
F087               104
F087               105 CLS0:                           ;画面消去ルーチン
F087 1E 11         106         LD      E,Y_WIDTH
F089               107 CLS0_LP00:
F089 60            108         LD      H,B
F08A 69            109         LD      L,C
F08B               110 CLS0_LP00':
F08B 16 12         111         LD      D,X_WIDTH
F08D AF            112         XOR     A
F08E               113 CLS0_LP01:
F08E ED 79         114         OUT     (C),A
F090 03            115         INC     BC
F091 15            116         DEC     D
F092 C2 8E F0      117         JP      NZ,CLS0_LP01
F095 44            118         LD      B,H
F096 4D            119         LD      C,L
F097 78            120         LD      A,B             ;次のﾗｲﾝへ
F098 C6 08         121         ADD     A,8
F09A 07            122         RLCA
F09B DA A3 F0      123         JP      C,NEXT_L
F09E 0F            124         RRCA
F09F 47            125         LD      B,A
F0A0 C3 89 F0      126         JP      CLS0_LP00
F0A3               127 NEXT_L:
F0A3 01 D8 37      128         LD      BC,37D8H
F0A6 B7            129         OR      A
F0A7 ED 42         130         SBC     HL,BC
F0A9 44            131         LD      B,H
F0AA 4D            132         LD      C,L
F0AB 1D            133         DEC     E
F0AC C2 8B F0      134         JP      NZ,CLS0_LP00'
F0AF C9            135         RET
F0B0 00            136 PG_FLG:         DS      1
```

リスト7 DEMO RUN.BAS

```
10 '      DEMO RUN.BAS
20 '
30 '      BY        Z.N
40 '
50 WIDTH 40
60 IF MEM$(&HE000,4)<>HEXCHR$("DD 2A 00 C2") THEN
PRINT"ｺｳｿｸ ﾗｲﾝ ﾙｰﾁﾝ (E000H ﾊﾞﾝ)ｶﾞ ｸﾐｺﾏﾚﾃｲﾏｾﾝ!":END
70 LOADM "DEMO2.EXE"
80 LOADM "DATA.OBJ",&HAE00
90 SCREEN 0,0:CLS 4:SCREEN 1,1:CLS 4
100 PALET 0,7:PALET 1,0
110 CALL &HF000
```

リスト8 DEMO DATA MAKE.BAS

```
10 '
20 '  IWAI SAN NO KAO WO GURUGURU MAWA SU.
30 '
40 '      PROGRAMMED BY Z.NISHIKAWA
50 '
60 'SAVE"ﾃﾞﾓ DATA MAKE.BAS"
70 DEFINT A,I:CLEAR &HC900:AD=&HC900:R=(3.1415926535#/180):B=.1
80 FOR I=0 TO 350 STEP 10
90 D=R*I:B=B+.025
100 RESTORE:GOSUB 170
110 AD=AD+2
120 NEXT
130 MEM$(AD-2,2)=MKI$(65533!)
140 'SAVEM"DATA.OBJ",&HC900,&HFAFF
150 END
160 '----- CALCULATION -----
170 READ X,Y
180 IF X=65534! THEN MEM$(AD,2)=MKI$(X):AD=AD+2:X=Y:READ Y
190 IF X=65535! THEN MEM$(AD,2)=MKI$(X):PRINT I;HEX$(AD+1):RETUR
N
200 X=(X-294)/2:Y=Y-100
210 X1=(X*COS(D)-Y*SIN(D))*B
220 Y1=(X*SIN(D)+Y*COS(D))*B
230 X=X1+294/2:Y=Y1+100
240 POKE AD,Y:MEM$(AD+1,2)=MKI$(X):AD=AD+3
250 GOTO 170
260 '----- DATA -----
270 DATA 269,89,272,89,272,90,269,90,65534,314,89,317,89,317,90,
314,90,65534,283,99,287,98,294,97,302,98,65534
280 DATA 304,99,305,100,308,101,305,105,293,106
290 DATA 284,105,279,103,280,100,65534,283,130,304,130,65534,364
,64,369,77,366,89,362,99
300 DATA 365,104,367,113,363,121,349,131,328,137,298,141
310 DATA 287,141,257,137,65534,255,136,239,131,226,123,223,116,2
25,107,228,101,222,85,220,78,221,72,226,65,65534
320 DATA 306,81,312,77,323,75,333,77,338,80,65534,281,81,276,77,
267,74,260,74,251,77,247,81,65534,239,56,248,51,271,45,291,44,29
9,44,326,48,345,54,65534
330 DATA 263,64,281,75,290,66,303,70,310,64,65534,328,68,345,74,
65534,335,69,346,65,65534,341,71,347,71,65534,346,58,356,54,368,
51,65534
340 DATA 356,62,370,61,65534,234,63,218,62,65534,239,59,225,55,2
18,54,65534,230,77,242,71,65534,235,70,231,68,65534,232,74,226,7
2,65534
350 DATA 369,86,378,85,387,87,391,93,385,102,375,105,367,106,655
34,222,88,208,87,201,89,198,94,204,103,214,107,223,108,65535,255
```

▶友人の家で新谷かおるの「ガッデム」を読んでいたら急にSprit of Rallyをやりたくなった。というわけで，さっさと入力しちまおうと思う今日このごろである。やっぱりラリーはいいなぁ～。
佐藤　史彦（16）北海道

特集1

——国際山岳救助隊——

パズルゲームThe Rescuer

Kamon Masato
華門 真人

プログラミングテクニックだけではプログラムは作れません。ここではプログラミングに至るまでの構想から設計段階までの過程を，パズルゲームの作り方を中心に解説してみましょう。単に遊ぶだけではいけません。これからはあなた自身の実践編が始まるのです。

パズルゲームとは

救助隊員(レスキュアー)，影の立役者。日陰の存在でありながら，事があったときにはもっとも頼りになる存在。今月はそんな彼らにヒントを得て，パズルゲームを作ってみた。

パズルゲームとひと口にいってもイロイロなものがあるが，一般的にはプログラミングは楽である（アクションゲームなんかと比べてのことであるが)。なぜか。それは絶対的なスピードを要求されないからである。アクションみたいに主人公がビュンビュン動き回る必要なんてないからね。

じゃあパズルゲームのプログラムを作るのはアクションゲームより簡単かというと，そうでもない。なぜなら，プログラミング自体は楽でも，ルールを決めたり，面を作ったりするのが大変だから。

つまりアクションはプログラミングテクニックに重点が置かれるのに対して，パズルでは発想力やそれをうまくプログラムに置き換える能力が問われることになる。いうならばアクションはエンジニア，パズルはプロデューサーみたいなものか。もっとも最近ではジャンル分けも難しいものが多いから，こう一概にはいいきれないだろうけれど。

まあ，結局どっちも，ひとつのプログラムとして作りあげるにはそれなりの根気を必要とするわけだ。

実際にパズルゲームを作ることを考えてみよう。パズルゲームの場合，まずなによりも先にルールを考えるという作業がくる。アクションの場合はルールなんて，どのゲームでもあまり大差ない。主人公を動かして，敵をやっつけるのだ。これはゼビウスでもアフターバーナーでも基本的に同じ。

ところが，パズルの場合，このルールそのものによってゲームが大きく左右される。というよりもルールがゲームのすべてといってしまっても過言ではない。ルール次第でゲームは面白くもなるし，まったくつまらないということにもなりうる。

ともかく，ルールを考える。で，できあがったらそれをプログラムへと置き換える。プログラムにするうちにルールに矛盾なども見えてくるであろうから，そういうものはうまくアレンジしてやる。パソコンには処理が重すぎる場合なんかもうまくアレンジして，なんとか動くようにする。

うまく動くようになった頃には，当初とは随分変わったルールになっていることだろう。が，コンピュータ上でちゃんとプログラミングされて動くルールこそが大事なのだ。頭の中にあるルールなどなんの役にも立ちはしない。

そうこうしてプログラムとして動くようになったら，プログラミングに関していえば9割方できあがりといえる。ところが1本のゲームとして見た場合は，まだ半分も完成してないことになる。

まだ肝心の「面」ができていないのだ。どんなに独特なルール，どんなに面白い発想も，それを十分に生かしきれる場，「面」を持たなければ，所詮，宝の持ち腐れにすぎない。パズルゲームでは，しっかりしたルールと，それを十分に生かした「面」がほとんどすべてといってもよい。プログラミングはそのあとからついてくるものだ。

もうちょっとつけ加えると，パズルゲームはルールが骨格としてあるから，プログラムとしては比較的綺麗なものになるはず（汚いプログラムで処理速度を上げる必要もないしね)。

もっとも，今回の僕のプログラムはお世辞にも綺麗とはいえないような気がする。う～ん，まあBASICのせいにしてしまうというのが無難かな（こらこら，自分の未熟さを棚にあげおって)。こういうタイプのプログラムはPASCALなんかだと整然と書けるはずなんだが。

一応言い訳をしておくと，究極の選択ではないけれど，綺麗だけれども動かないプログラムと，汚いけれども動くプログラムでは後者のほうがはるかにましである。

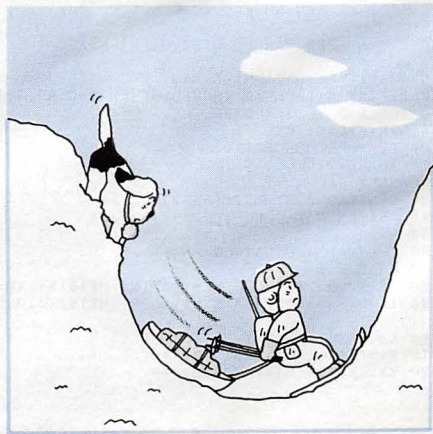
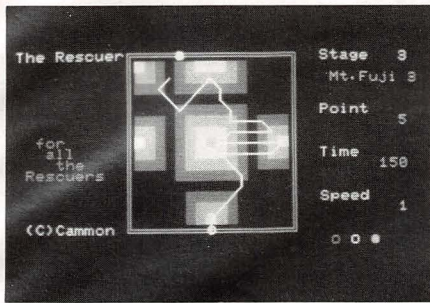
あともうひとつ，先ほど僕は面が重要と書いたが，これには逃げ道がある。つまり制限を加えてやることだ。すなわち，プレイヤーに十分に考えられるだけの時間を与えないように，時間制限をつけたり，敵を作ったりするのだ。

邪道のような気がするが，これはフラッピーやテトリスなんかでも使われている手だ。まあ許してもらおう。制限をつけない，たとえば倉庫番（懐かしい！）みたいなプログラムは，面がよっぽど奥の深いものでないと面白くならないからね。

いずれにせよ，パズルゲームは発想（ルール，面)が第一，あとはその発想をプログラムに置き換えていけばよい。それゆえ，プログラミング入門としても最適のような気がするんだけど，いかがだろう。

レスキュアー

レスキュアー，救助隊員。彼らは決して目立つことのない存在だ。山岳救助隊ならなおさらだ。これがまだ海難救助なんかだと，真っ黒に日焼けしていて海辺でもひときわ目立つんだが。

スキーシーズンの山は華やかだ。山岳救助隊員がいかにスキーがうまくても所詮，まわりに埋もれてしまう。が，ひとたび救助の要請があれば，彼らが主役だ。一刻も

早く助けて，安全なところまで運ぶ。

ときにはバカなスキーヤーのために命まで賭ける。そこには名声などを追い求めない真摯な姿がある。

そんなイメージでゲームを作ってみよう。もっともこれだけではゲームにするにはちょっと足りない。少し想像の翼を広げてみよう。J・アーヴィングもいうように，想像力が必要なのだ。

近い将来，摩擦のまったくない物質が発明されたとしよう。とするとどうなるか。スキーでいうならば，滑り降りた分だけまた登れるのだ（もうリフトなんてものはいらなくなる？)。もっともどうやってスキーをコントロールするんだろうね。

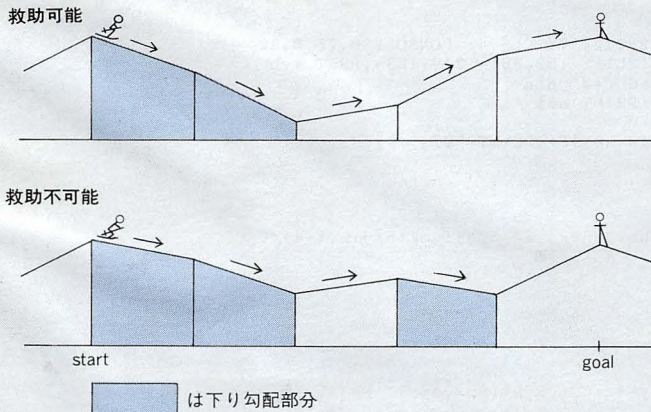
さらに想像の翼を広げる。もしも山岳救助隊員がこの摩擦のまったくない物質を使ったら？　ここらへんでようやくゲームらしきものがその姿を現す。

これをちゃんとしたルールに置き換えなければならない。すなわち，

滑り降りた分だけ登れる

もっともこれではまだコンピュータに落とすには漠然としているが。

ルール

上で述べたような発想を基にプログラムに仕立てたのがリスト2だ。

ルールを述べよう。まず根本には「滑った分だけ登ってこれる」というものがある。さらにこれをゲームとして仕立てるために次のようなルールを作った。

- 滑り降りるとスピードがひとつ増え，登るとひとつ減る。
- ただし，山の頂上や人を救助する場合などは自動的にスピードが0になる。
- 同じ高さのところへは進めない。登るか，降りるかである。
- 丘を越えるとスピードが落ちる。移動は降下，登坂をひと組にして行う。
- 進めるのは前方・右前方・左前方の3方向のみ。ただし，スピードが0のときにはどの方向にでも行ける。
- 同じところは二度は通れない。まあ，摩擦のない物質のレールでも引いているのだと考えましょう。

この際ルールなんてはっきりいって非現実的でかまわない。よくゲームをやっていて「こんなことがあるはずがない」という人がいるけれど，それは間違い。そのゲームの世界では本当にそういうことが起こるのだ。ゲームこそが世界であり，すべてなのだから，現実に縛られる必要はまったくない。現実に縛られていたらせせこましいじゃないねェ。

村上春樹の世界ではVWビートルにラジエターがついているように，ゲームの世界ではその世界独特の法則が働いているのだ。

操作方法

実際の操作は簡単である。テンキー（1～9）を使って行きたい方向に進む。もちろん上のルールを守らないと進めないが。注意したいのは進める方向である。スピードが出ていると，3方向にしか進めない。それゆえどうしても方向を変えるときには5キーでブレーキ（スピードがひとつ下がる）をかけてスピードを0にしてから曲がらねばならない。

また操作は機敏にしないと時間切れになってしまう。考えているヒマはないゾ。

高度は色で表現されている。すなわち，黒は高度0，青は高度1，赤は2，そう，パレットコードがそのまま高度になるのだ。

目的は「○」のところから出発して点滅している遭難者を助けにいき，「●」のところまで運ぶことだ。遭難者はひとりとは限らないし，また人間とも限らない？

とまあルールは複雑そうだがいっぺんやればだいたいつかめるだろう。ひとりでも多くの人を助けるのが使命だ。

プログラム

参考までにプログラムについてもちょっと触れておこう。メインルーチンは1820行から2210行までだ。この中に上で述べたルールがプログラムとして収まっている。なんならどのルールが何行目にあるかを調べてみると勉強になるかもしれない。

そして2720行以降が面のデータだ。これだけで15面分あることになる。このデータは圧縮されている（1キャラクタ3ビット）ので，内容を読むのは難しいかもしれない。展開したければ1630行あたりを参考にしてほしい。

また，自分で面を作りたいという人はリスト1を利用するといいだろう。これは高度（0～6の数字）を入力すると，圧縮したデータを生成してくれる。こうして得られたデータに遭難者の数，場所（X，Y座標），出発地点（X，Y），目的地点（X，Y）を加えてDATA文とすればOK。

できることならば，あらかじめ用意された15面以外に，自分で面を作って遊んでみてほしい。さらに面白い面ができあがった

ひたすら降りてひたすら昇る

リスト1

```
1000 'The Rescuer Construction Kit
1010 '
1015 DEFINT a-z: DIM mdt(15,3)
1020 FOR i=1 TO 15
1030 PRINT i;"line:";: INPUT "",in$
1035 FOR j=1 TO 3
1040 FOR k=0 TO 4
1045 dt1=VAL(MID$(in$,k+1+(j-1)*5,1))
1050 dt=dt+2^(k*3)*dt1
1060 NEXT
1070 mdt(i,j)=dt: dt=0
1080 NEXT
1090 NEXT
1100 FOR i=1 TO 15
1110 FOR j=1 TO 3
1120 PRINT mdt(i,j);",";
1130 NEXT
1140 PRINT ""
1150 NEXT
1160 END
```

らぜひ投稿してほしい。いつか日の目を見るかもしれないしね。

パズルゲームについて その2

こうしてパズルゲームができあがるまでをだいぶ駆け足でたどってみたわけだが，どうだろう。アイデアだったら自信があるという君も挑戦してみてはいかがかな。

何度もいっているように，パズルゲームではプログラミングテクニックはそれほど必要ではない。むしろアイデアのほうが重要だ。まずアイデアを固めることから始めてみよう。そしてそれを徐々にプログラムにする。不都合な点は適当に調整する。

時間はかかるかもしれない。だが，一度できあがれば，それは必ずそこそこに楽しめるものになっているはずだ。アイデアさえしっかりしていれば。

問題は技術よりも発想だ。ぜひ一度トライしてみてほしい，想像の翼を広げて。

リスト2

```
1000 '*********************************************
1010 '
1020 '     for X1/X1turbo
1030 '
1040 '                'The Rescuer'
1050 '
1060 '                          (C) Cammon 1989,8
1070 '
1080 '*********************************************
1090 '
1100 INIT: CLS 4: PRW 255
1110 WIDTH 40                                                 'for X1
1120 'WIDTH 40,25,0,2: KMODE 0: KLIST 0                       'for X1turbo
1130 CONSOLE 0,25: CLICK OFF: DEFINT a-z: KBUF OFF
1140 DIM ht(15,15),ck(15,15),pt$(9,11),cr$(9): stn=1
1150 ps$(1)="man": ps$(2)="woman": ps$(3)="pig"
1160 ms$(1)="     Time out!     ": ms$(2)=" Unapproachable!"
1170 ms$(3)="   Unfinished!   "
1180 dr(0)=8: dr(1)=1: dr(2)=2: dr(3)=3: dr(4)=4: dr(5)=5: dr(6)=6
1190 dr(7)=7: dr(8)=8: dr(9)=1
1200 '
1210 'Make initial screen
1220 SCREEN 1,0: LINE (90,34)-(221,165),PSET,4,b
1230 LINE (92,36)-(219,163),PSET,4,b
1240 LINE (93,37)-(218,162),PSET,2,bf: PALET 2,0: SCREEN 0,0: PRW 128
1250 FOR i=1 TO 9
1260  READ cr$(i),ptn$
1270  FOR j=1 TO 11
1280   ptdt=VAL("&H"+MID$(ptn$,j*4-3,4))
1290    FOR k=1 TO 13
1300     exst=ptdt MOD 2: ptdt=INT(ptdt/2)
1310     IF exst THEN pt$(i,j)=cr$(i)+pt$(i,j) ELSE pt$(i,j)=" "+pt$(i,j)
1320    NEXT
1330   NEXT
1340 NEXT
1350 COLOR 5
1360 FOR i=1 TO 11
1370  READ zyn: IF fl=0 THEN GOSUB 2240: BEEP
1380  LOCATE i-1,4: PRINT cr$(zyn)
1390  IF INKEY$<>"" THEN fl=1
1400 NEXT: PAUSE 10
1410 FOR i=1 TO 9
1420  FOR j=1 TO 11
1430   pt$(i,j)=""
1440  NEXT
1450 NEXT
1460 SCREEN 1,0: PRW 255: LINE (93,37)-(218,162),PSET,0,bf: PALET
1470 COLOR 2: LOCATE 2,12: PRINT "for"+CHR$(&HD)+"  all"+CHR$(&HD)+"   the"
1480 PRINT " Rescuers": COLOR 5: LOCATE 1,20: PRINT "(C)Cammon"
1490 LOCATE 30,4: PRINT "Stage"
1500 LOCATE 30,9: PRINT "Point"
1510 LOCATE 30,13: PRINT "Time"
1520 LOCATE 30,17: PRINT "Speed"
1530 LOCATE 31,21: COLOR 2: PRINT "● ";: COLOR 6: PRINT "○ ";
1540 COLOR 4: PRINT "○"
1550 '
1560 'draw stage
1570 SCREEN 1,0: PRW 255: CONSOLE 0,25,12,17: CLS 4: CONSOLE 0,25,0,40
1580 LINE (90,34)-(221,165),PSET,4,b: LINE (92,36)-(219,163),PSET,4,b
1590 COLOR 5: LOCATE 37,4: PRINT USING "##",stn
1600 COLOR 4: READ nm$: LOCATE 31,6: PRINT nm$
1610 LOCATE 38,10: PRINT "0"
1620 LOCATE 36,14: PRINT "200": LOCATE 38,18: PRINT "0"
1630 FOR i=5 TO 19
1640  FOR j=0 TO 2
1650   READ sd
1660   FOR k=1 TO 5
1670    sd1=sd MOD 8: sd=INT(sd/8): ht(5*j+k,i-4)=sd1: ck(5*j+k,i-4)=0
1680    LOCATE 11+5*j+k,i: COLOR sd1: PRINT "■"
1690   NEXT
1700  NEXT
1710 NEXT
1720 '
1730 READ spn: CFLASH 1
1740 FOR i=1 TO spn
1750  READ spx,spy: LOCATE spx+11,spy+4: COLOR ht(spx,spy): PRINT "■"
1760  sx(i)=spx: sy(i)=spy
```

```
1770 NEXT: CFLASH
1780 COLOR 7: READ stx,sty: LOCATE stx+11,sty+4: PRINT "○"
1790 READ qtx,qty: LOCATE qtx+11,qty+4: PRINT "●"
1800 '
1810 'play
1820 SCREEN 0,0: PRW 128: odir=5: spd=0
1830 LINE (91+8*stx,36+8*sty)-(91+8*stx,28+8*sty),PSET,7: px=stx: py=sty-1
1840 GOSUB 2410
1850 REPEAT
1860  tm=200: ck(px,py)=1: opx=px: opy=py
1870  in$=INKEY$: IF in$=CHR$(&H1B) THEN 2510
1880  ON VAL(in$) GOTO 2080,2090,2100,2110,2120,2130,2140,2150,2160
1890  tm=tm-1: LOCATE 36,14: COLOR 4: PRINT USING "###",tm
1900  IF tm=0 THEN ms1=1: GOTO 2450 ELSE 1870
1910  IF px<1 OR px>15 THEN ms1=2: GOTO 2450
1920  IF py<1 OR py>15 THEN ms1=2: GOTO 2450
1930  IF ck(px,py) THEN ms1=2: GOTO 2450
1940  IF er THEN ms1=2: GOTO 2450
1950  LINE (91+8*opx,36+8*opy)-(91+8*px,36+8*py),PSET,7
1960  FOR i=1 TO spn
1970   IF px=sx(i) AND py=sy(i) THEN GOSUB 2330
1980  NEXT
1990  LOCATE 38,18: COLOR 4: PRINT USING "#",spd
2000 UNTIL (px=qtx) AND (py=qty+1)
2010 LINE (91+8*qtx,44+8*qty)-(91+8*qtx,36+8*qty),PSET,7
2020 IF spn THEN ms1=3: GOTO 2450
2030 BEEP: BEEP: LOCATE 13,22: CFLASH 1: COLOR 2: PRINT "You Made it !": CFLASH
2040 PAUSE 70: LOCATE 13,22: PRINT CHR$(&H1A)
2050 stn=stn+1: IF stn<16 THEN 1570 ELSE GOTO 2510
2060 '
2070 'judge
2080 dir=2: px=px-1: py=py+1: GOTO 2170
2090 dir=1: py=py+1: GOTO 2170
2100 dir=8: px=px+1: py=py+1: GOTO 2170
2110 dir=3  px=px-1: GOTO 2170
2120 IF spd THEN spd=spd-1: GOTO 1990 ELSE GOTO 1890
2130 dir=7: px=px+1: GOTO 2170
2140 dir=4: px=px-1: py=py-1: GOTO 2170
2150 dir=5: py=py-1: GOTO 2170
2160 dir=6: px=px+1: py=py-1
2170 IF ht(px,py)=ht(opx,opy) THEN er=1
2180 IF spd>0 AND dir<>odir AND dir<>dr(odir+1) AND dir<>dr(odir-1) THEN er=1
2190 IF ht(px,py)<ht(opx,opy) THEN IF ud THEN spd=1: ud=0 ELSE spd=spd+1
2200 IF ht(px,py)>ht(opx,opy) THEN ud=1: IF spd THEN spd=spd-1 ELSE er=1
2210 odir=dir: GOTO 1910
2220 '
2230 'draw character patterns
2240 PRW 132
2250 FOR j=1 TO 11
2260  LOCATE 13,j+7
2270  PRINT pt$(zyn,j)
2280 NEXT: PRW 128
2290 PAUSE 5: CONSOLE 0,25,12,28: CLS: CONSOLE 0,25,0,40
2300 RETURN
2310 '
2320 'Rescue
2330 BEEP: LOCATE px+11,py+4: COLOR ht(px,py): PRINT "■"
2340 sx(i)=sx(spn): sy(i)=sy(spn): i=spn: spn=spn-1: svp=INT(RND*3)+1
2350 LOCATE 10,22: COLOR 5: PRINT "Ok! You save a ";ps$(svp);"."
2360 pot=pot+1: COLOR 4: LOCATE 36,10: PRINT USING "###",pot: PAUSE 5
2370 GOSUB 2410: LOCATE 10,22: PRINT CHR$(&H1A)
2380 pot(svp)=pot(svp)+1: spd=0: RETURN
2390 '
2400 'Signal
2410 COLOR 4: LOCATE 35,21: PRINT "○"
2420 BEEP: COLOR 2: LOCATE 31,21: PRINT "●": PAUSE 12
2430 BEEP: LOCATE 31,21: PRINT "○";: COLOR 6: PRINT " ●": PAUSE 12
2440 BEEP: LOCATE 33,21: PRINT "○";: COLOR 4: PRINT " ●": RETURN
2450 '
2460 'Miss
2470 LOCATE 12,22: COLOR 2: CFLASH 1: PRINT ms$(ms1): CFLASH
2480 LOCATE 15,24: PRINT "You Failed!";: PAUSE 70
2490 '
2500 'Result
2510 CLS 4: COLOR 5: LOCATE 5,5: PRINT "Well,you saved"
2520 LOCATE 13,8: PRINT USING "##",pot(1);: PRINT "  men"
2530 LOCATE 13,10: PRINT USING "##",pot(2);: PRINT "  women"
2540 LOCATE 13,12: PRINT USING "##",pot(3);: PRINT "  pigs"
2550 LOCATE 15,20: PRINT "Hit any key...";: in$=INKEY$(1)
2560 IF in$=CHR$(27) THEN 2610
2570 stn=1: fl=1: pot=0: pot(1)=0: pot(2)=0: pot(3)=0: er=0
2580 CLS: RESTORE 2710: PRW 255: GOTO 1350
2590 '
2600 'Break
2610 CLS 4: INIT: WIDTH 80: END
2620 DATA T,1fff0040004000400040004000400040004000400040
2630 DATA h,020002000200020002e003100208020802080208 0208
2640 DATA e,000000000000000000e00110020803f80200011000e0
2650 DATA " ",00000000000000000000000000000000000000000000
2660 DATA R,07f00408040404040408 07f00480044004200410040c
2670 DATA s,000000000000000001f00208020001f00008020801f0
2680 DATA c,000000000000000001f00208020002000200020801f0
2690 DATA u,00000000000000000208020802080208020802 1801e8
```

```
2700 DATA r,00000000000000000230024802800300020002000200
2710 DATA 1,2,3,4,5,3,6,7,8,3,9
2720 DATA Mt.Fuji  ,4179,9946,1,4178,9362,1,4169,4681,1,0
2730 DATA 0,0,4096,4681,1,4169,9362,4673,4178,9946
2740 DATA 9281,4179,10010,13377,4178,9946,9281,4169,9362,4673
2750 DATA 4096,4681,1,0,0,0,0,4681,0,0
2760 DATA 5265,0,0,5329,0
2770 DATA 1,1,8,8,16,8,0
2780 DATA Mt.Fuji 2,4179,9946,1,4178,9362,1,4169,4681,1,0
2790 DATA 0,0,4096,4681,1,4169,9362,4673,4178,9946
2800 DATA 9281,4179,10010,13377,4178,9946,9281,4169,9362,4673
2810 DATA 4096,4681,1,0,0,0,0,4681,0,0
2820 DATA 5265,0,0,5329,0
2830 DATA 2,8,7,1,8,8,16,8,0
2840 DATA Mt.Fuji 3,4179,9946,1,4178,9362,1,4169,4681,1,0
2850 DATA 0,0,4096,4681,1,4169,9362,4673,4178,9946
2860 DATA 9281,4179,10010,13377,4178,9946,9281,4169,9362,4673
2870 DATA 4096,4681,1,0,0,0,0,4681,0,0
2880 DATA 5265,0,0,5329,0
2890 DATA 2,9,8,7,3,8,16,5,0
2900 DATA Taweche  ,9937,9937,13377,9937,9361,9281,9937,4681,4673,9361
2910 DATA 1,0,4681,4673,73,4096,9281,82,4617,9281
2920 DATA 4179,10,13376,82,4618,9281,82,8714,4673,73
2930 DATA 4617,1,0,4096,4681,4673,4169,9362,9281,4178
2940 DATA 9946,13377,4179,10010,13377
2950 DATA 1,4,2,1,16,15,0
2960 DATA Taweche 2,9937,9937,13377,9937,9361,9281,9937,4681,4673,9361
2970 DATA 1,0,4681,4673,73,4096,9281,82,4617,9281
2980 DATA 4179,10,13376,82,4618,9281,82,8714,4673,73
2990 DATA 4617,1,0,4096,4681,4673,4169,9362,9281,4178
3000 DATA 9946,13377,4179,10010,13377
3010 DATA 1,15,14,1,16,13,0
3020 DATA Taweche 3,9937,9937,13377,9937,9361,9281,9937,4681,4673,9361
3030 DATA 1,0,4681,4673,73,4096,9281,82,4617,9281
3040 DATA 4179,10,13376,82,4618,9281,82,8714,4673,73
3050 DATA 4617,1,0,4096,4681,4673,4169,9362,9281,4178
3060 DATA 9946,13377,4179,10010,13377
3070 DATA 3,5,10,4,4,11,3,1,16,13,0
3080 DATA Lhotse   ,667,4177,4681,658,73,0,585,0,4681,0
3090 DATA 584,9361,4680,585,4681,9352,82,8704,13960,4179
3100 DATA 4617,18056,4179,8714,13960,4179,4617,9352,82,0
3110 DATA 4680,4169,4680,0,4096,9352,4681,4680,9864,5265
3120 DATA 648,9864,5329,648,9865
3130 DATA 1,5,5,9,16,6,0
3140 DATA Lhotse 2 ,667,4177,4681,658,73,0,585,0,4681,0
3150 DATA 584,9361,4680,585,4681,9352,82,8704,13960,4179
3160 DATA 4617,18056,4179,8714,13960,4179,4617,9352,82,0
3170 DATA 4680,4169,4680,0,4096,9352,4681,4680,9864,5265
3180 DATA 648,9864,5329,648,9865
3190 DATA 1,14,12,3,16,8,0
3200 DATA Lhotse 3 ,667,4177,4681,658,73,0,585,0,4681,0
3210 DATA 584,9361,4680,585,4681,9352,82,8704,13960,4179
3220 DATA 4617,18056,4179,8714,13960,4179,4617,9352,82,0
3230 DATA 4680,4169,4680,0,4096,9352,4681,4680,9864,5265
3240 DATA 648,9864,5329,648,9865
3250 DATA 2,6,10,3,2,8,16,7,0
3260 DATA Pikes    ,9362,9362,13377,4681,8777,9281,0,4608,4673,4681
3270 DATA 9,0,9362,4618,585,4682,8713,658,10,4608
3280 DATA 585,4618,585,0,8714,594,4681,8713,658,9361
3290 DATA 4608,585,14033,9,0,14033,4682,584,9361,5266
3300 DATA 648,4681,5339,648,0
3310 DATA 2,7,13,12,13,3,16,13,0
3320 DATA Pikes 2  ,9362,9362,13377,4681,8777,9281,0,4608,4673,4681
3330 DATA 9,0,9362,4618,585,4682,8713,658,10,4608
3340 DATA 585,4618,585,0,8714,594,4681,8713,658,9361
3350 DATA 4608,585,14033,9,0,14033,4682,584,9361,5266
3360 DATA 648,4681,5339,648,0
3370 DATA 2,5,5,13,6,3,16,14,0
3380 DATA Pikes 3  ,9362,9362,13377,4681,8777,9281,0,4608,4673,4681
3390 DATA 9,0,9362,4618,585,4682,8713,658,10,4608
3400 DATA 585,4618,585,0,8714,594,4681,8713,658,9361
3410 DATA 4608,585,14033,9,0,14033,4682,584,9361,5266
3420 DATA 648,4681,5339,648,0
3430 DATA 2,7,9,14,2,3,16,3,0
3440 DATA Jungfrau ,5339,18202,18724,5347,14106,14043,5339,9946,9362,5266
3450 DATA 9362,4681,4681,4681,9361,5266,0,14033,5339,0
3460 DATA 18641,5347,0,14033,5339,0,9361,5266,0,4681
3470 DATA 4681,4681,9361,9362,8850,14034,14043,8859,18651,18724
3480 DATA 8860,14563,23405,8860,18659
3490 DATA 1,10,1,1,16,1,0
3500 DATA Jungfrau2,5339,18202,18724,5347,14106,14043,5339,9946,9362,5266
3510 DATA 9362,4681,4681,4681,9361,5266,0,14033,5339,0
3520 DATA 18641,5347,0,14033,5339,0,9361,5266,0,4681
3530 DATA 4681,4681,9361,9362,8850,14034,14043,8859,18651,18724
3540 DATA 8860,14563,23405,8860,18659
3550 DATA 1,12,14,4,16,10,0
3560 DATA Jungfrau3,5339,18202,18724,5347,14106,14043,5339,9946,9362,5266
3570 DATA 9362,4681,4681,4681,9361,5266,0,14033,5339,0
3580 DATA 18641,5347,0,14033,5339,0,9361,5266,0,4681
3590 DATA 4681,4681,9361,9362,8850,14034,14043,8859,18651,18724
3600 DATA 8860,14563,23405,8860,18659
3610 DATA 2,3,3,14,7,5,16,7,0
```

文字列操作の基本

Murata Toshiyuki 村田 敏幸

コンピュータ内部での“文字”の扱われ方や操作の仕方。今月の課題はこれです。ファイル処理への前段階ともいえる文字列処理。新しい命令やサンプルプログラムなども次々と出てきてこなしがいがありますが，焦らずていねいに読んでいってください。

今回は文字列操作について話してみる。これは次回以降やる予定のファイル処理の準備段階にあたる。

はっきりいって，ファイル処理はマシン語っぽくない地味な処理だ。その割には結構入り組んでいて“正しく”作るのは思いのほか骨が折れる。その前提である文字列の処理も，マシン語で書くと煩雑になってしまう場合が多い。ファイル処理・文字列処理はマシン語が一番苦手な分野といえるかもしれない。

なのに敢えてやるからには，それなりの見返りがあるはずなのだが……具体的にどんな“いいこと”があるかはちょっと説明できそうもない。読者が自力で見つけてくれることを期待したい。

文字列の表現法

コンピュータ内部では“文字”はその文字コードという数字の形で表現される。“文字列”は言葉どおり文字が列になったものだから，文字コードの列によって表されるだろう。

たとえば，“ABC”という文字列は，任意のアドレス（かりに80000H番地）から，“A”，“B”，“C”の文字コード（半角文字であればASCIIコード）を順に並べることで表現できる。

80000H ← 41H(A)
80001H ← 42H(B)
80002H ← 43H(C)

実際には，この形では文字列の大事な属性である“長さ（何文字の文字列か）”という情報が欠けている。文字列がどこで終わるかがわからないのだ。

ストレートな発想をするなら，先頭の1バイトに文字列の長さを格納しておけば，一応の格好はつく。

80000H ← 03H（文字列の長さ＝3）
80001H ← 41H（A）
80002H ← 42H（B）
80003H ← 43H（C）

というようにだ。これにより，先頭の1バイトを見れば文字列の長さがすぐ得られ，また文字列の終端がどこかを知ることができるようになる。

しかしこの方法には，一定以上の長さの文字列を表現できないという欠点がある。つまり，1バイトで表すことができる数（0～255）によって，文字列の長さが制限されてしまうのだ。

文字列を表現する別の考え方としては，ある決まったコードで文字列の終端を表すという方法がある。仮に“/”で文字列の終わりを表現することにすれば上の例は，

80000H ← 41H（A）
80001H ← 42H（B）

一般のBASICではこの形に似た形式で文字列を表現するので，文字列は255文字までという制限がある。

OSや言語によっては，この終了コードは“$”だったり改行コード（0DH)だったりすることもある。Cは00Hで文字列の終わりを表す方式を採っている。ちなみに00Hで終わる形式の文字列のことを“ASCIIZ文字列”とか，縮めて“ASCIZ文字列”と呼ぶことがある。

.dc疑似命令

忘れられていると困るので念のため書いておくと，疑似命令というのは68000の命令ではなくアセンブラに対する指令だ。ラベルに値を定義するのに使うequや，任意の大きさのメモリ領域を確保する.ds，そして，データを用意する.dcなどはみんな疑似命令だ。

.dsがメモリを確保するだけなのに対し，.dcはメモリを確保して，そこを指定の定数で初期化する。例によって，.b，.w，.lのバリエーションがあり，それぞれバイト長，ワード長，ロングワード長のデータを用意するのに使う。

たとえば，

```
.dc.b    10
```

は1バイトの領域を確保し，そこに10を入れておくことを意味する。いいかえると，10という値をそのままオブジェクトに埋め込むことになる。極端な話，アセンブリ言語の命令がどのようなコードに変換されるか知っていれば，.dcだけを使ってプログラムを書くことができる。もっともそんなことをわざわざする人もいないだろうが。

.dcの後ろには複数のデータをカンマで区切って並べることができる。

```
.dc.b    10, 20, 30
```

なら，3バイトの連続したメモリ領域に10，20，30が順に書き込まれる。また，文字はクオーテーションマークでくくって表すことができるから，

```
.dc.b    'A', 'B', 'C'
```

は，

```
.dc.b    $41, $42, $43
```

と同じ意味になるし，さらには，

```
.dc.b    'ABC'
```

のようにまとめて表現することもできる。これは文字列データを用意する際の基本形だ。

80002H ← 43H (C)

80003H ← 2FH (/)

という形になる。この場合，先頭から終了コード"/"までの文字数（バイト数）を数えることで，文字列の長さが得られる。

この形式ならば文字列の長さはいくらでも長くすることができる。ただ，終了コード自体を文字列の中に含むことができない（途中に終了コードを入れると，そこで文字列が終わってしまう）ので，終了コードを何にするかはうまく選ばなければならない。プログラムの中でつじつまがあっていれば何でもいいのだが，一般的なのは00Hを終了コードにする方法だ。これはプログラムを書く際にもなかなか便利であり，Human68kのDOSコールもそうなっているので，今後文字列はこの形で表現することにする。

ここで，メモリ上に置かれた文字列は常にその"先頭アドレス"によって指定することができることを覚えておいてもらいたい。たとえば，

80000H ← 41H (A)

80001H ← 42H (B)

80002H ← 43H (C)

80003H ← 00H (終了コード)

という文字列は，80000Hという先頭アドレスひとつで間接的に表すことができるだろう。

さて，リスト1-aはDOSコールprintを利用した文字列表示の例だ。文字列はラベルmes以下に.dc.bによって用意し，その先頭アドレスをスタックに積んでDOSコールを呼び出している。.dcは前にも簡単に紹介したが，この際コラム「.dc疑似命令」で，もう少し詳しい話をしておく。

文字列は00Hで終わるわけだから，改行コード0DHや0AHも文字列中に含めることができる点は見落とさないでもらいたい。必要とあらば，ひとつの文字列の中にいくつも改行コードをはさんで，1回の文字列表示で複数行の表示を行うことだってできるということだ。

ここで，今後のこともあってリスト1-aのメイン

リスト1 PRT.S

```
a)   1: *        DOSコールによる文字列表示
     2:
     3:          .include         doscall.mac
     4: *
     5:          .text
     6:          .even
     7: *
     8: ent:
     9:          lea.l   mysp,sp            *spの初期化
    10:
    11:          move.l  #str,-(sp)         *文字列先頭アドレスをプッシュ
    12:          DOS     _PRINT             *文字列表示
    13:          add.l   #4,sp              *スタック補正
    14:
    15:          DOS     _EXIT              *終了
    16: *
    17: str:     .dc.b   '1234ABCD'         *表示する文字列
    18:          .dc.b   $0d,$0a,0          *
    19: *
    20:          .stack
    21:          .even
    22: *
    23: mystack:
    24:          .ds.l   256                *スタック領域
    25: mysp:
    26:          .end

b)   8: ent:
     9:          lea.l   mysp,sp            *spの初期化
    10:
    11:          pea.l   str                *文字列先頭アドレスをプッシュ
    12:          DOS     _PRINT             *文字列表示
    13:          addq.l  #4,sp              *スタック補正
    14:
    15:          DOS     _EXIT              *終了
```

命令のバリエーション

move, add, sub, cmp, and, or, eorは厳密にはオペランドによっていくつかのバリエーションがある。ここら辺は68000のドロドロとした部分で，真面目に区別しようとすると悲惨なことになるかもしれない。

たとえば，

move.l　d0, a0
add.l　d0, a0
sub.l　d0, a0

という命令は本当はなく，正しくは，

movea.l　d0, a0
adda.l　d0, a0
suba.l　d0, a0

と，Addressの"a"をつけた命令を使わなければならない。また，

add.l　＃10, d0
sub.l　＃10, d0

は，

addi.l　＃10, d0
subi.l　＃10, d0

のようにImmediateの"i"をつけるのが正しい。

といって，

movei.l　＃10, d0
addi.l　＃10, a0

という命令はありそうで，ない。

さて，洞察力に優れた人なら，上に挙げた例から何らかの規則性を見つけることができるだろう。また，几帳面な人なら，『アセンブラマニュアル』か『68000プログラマーズ・ハンドブック』かなんかで各命令とアドレッシングモードの組み合わせの有効・無効を調べようと思ったかもしれない。

が，たぶん多くの読者は「めんどくせー」と思ったことだろう。その感覚は全面的に正しい。これはいってみれば，BASICのPRINT文に整数表示用，実数用，文字列用が別々に用意されているようなもので，使い分けるのはうざったいだけだ。

そこで，たいていの68000用アセンブラは，単にmoveと書けば，アセンブラが勝手に適切な命令に置き換えてくれるように作られている。だから読者は慌ててmoveaだのaddiだのの新しい命令を覚える必要はなく，これまでどおりデータ転送ならmove，加算ならadd，減算ならsub，論理演算ならand, or, eorを使えばよい。

とはいうものの，僕は運よくというか運悪くというか，命令を正しく使い分けるのが癖になってしまっている。今後のリストにはmoveaやsubiなんかがちょくちょく登場することになるだろう。読者に僕の癖を押しつけるつもりはないが，一応ここで話したことは頭に入れておいて，リスト中の"知らない命令"に驚かないようにしてもらいたい。

ただの潔癖性だと思われるのは癪なのでちょっと弁解しておくと，僕が命令をフルスペルで書くのには保険みたいな意味がないでもない。

たとえば，

add.l　＃10, d0

の"＃"をつけ忘れて

add.l　10, d0

と書いてしまうのはありがちな単純ミスだし，

move.l　d1, a0

は1カ所打ち間違うだけで，

move.l　d1, d0

に化けてしまうが，いつも

addi.l　＃10, d0
movea.l　d1, a0

と書くようにしていれば，万一ミスタイプしても，

addi.l　10, d0
movea.l　d1, d0

をエラーとしてアセンブラがはじいてくれる。

僕はドジなもんで，なるべくミスが早目に発見できるように予備工作しておかないと信頼性の高いプログラムが作れないんだよね。

ということころで，59ページのコラム「クイック・イミディエイト・アドレッシング」へと続く。

ルーチンを若干格好よく書き直してみた。リスト1-bだ。peaとaddqという新しい命令が顔を出している。これらに関してもコラム「命令のバリエーション」と「実効アドレス」を参照してもらいたい。

他言語にみる文字列操作

マシン語での文字列操作の仕方に入る前に，ほかの言語では文字列をどのように扱っているか確認しておく。お馴染みのX-BASICとCを取り上げよう。

X-BASICでは，

```
str  a, b, c
```

てな具合に文字列型の変数を用意しておけば，

```
a="ABCDEFG"
b=a
c=a+b
if(a=b)then~
```

のように代入・連結・比較などの処理が簡単に行える。見た目では数値変数を扱う場合と何も変わらないといっていい。さすがは高級言語というべきか。

C言語になると文字列型の変数というものはない。が，char型の配列を，

```
char a[256], b[256], c[256];
```

のように定義してから，あらかじめ用意された文字列操作関数を使い，

```
strcpy(a, "ABCDEFG");
strcpy(b, a);
strcat(strcpy(c, a), b);
if(!strcmp(a, b))~
```

という感じで同等の処理ができる。

ではマシン語はというと，文字変数なんて便利なものはないし，文字列操作用の命令が用意されているわけでもない。

ここで，Cにおける文字列の扱い方には参考になる部分も多い。

```
char a[256];
```

を“256バイトの領域を確保し，その先頭アドレスをaで表す”というように読みかえると，これはアセンブリ言語の，

```
a:  .ds.b   256
```

に直接対応することがわかる。aやbなどの配列名を“アドレスを表すラベル”だと思えば，

```
strcpy(b, a);
```

は，“a，bという2つのアドレスを指定して，strcpyというサブルーチンを呼び出す”ようなものだ。

こと文字列操作に関する限り，Cとマシン語の差はstrcpyなどの関数があらかじめ用意されているかどうかの違いしかないことになる。となると，Cの文字列操作関数に相当するサブルーチンを一度作っておけば，あとはそのサブルーチンを使い回しにすることでC程度には楽に文字列を扱えるようになるだろう。

次の節以降では，C風の文字列操作サブルーチンをいくつか作ってみよう。

文字列の複写

アドレスstr2から格納された文字列を，アドレスstr1以下にコピーすることを考える。文字列は 00_H で終わることにしてあるから，この処理は“アドレスstr1から 00_H に出会うまで，1バイトずつstr2以下に転送する”といいかえることができる。

連続したメモリ領域を操作するには“ポインタ”の考え方が有用だ。68000のマシン語でいうと“アドレスレジスタ間接アドレッシング”を利用することになる。

まず，2つのメモリ領域の先頭アドレスをアドレスレジスタに入れておく。a0に転送先，a1に転送元を入れることにすると，

```
lea.l str1, a0
```

実効アドレス

混乱しないように気をつけて読んでもらいたい。

```
move.l    LABEL, d0
```

は“LABELというラベルで指定されるアドレスからのロングワードデータをd0に転送する”という意味だが，ここで“LABELで指定されるアドレス”のことを実効アドレス（Effective Address）と呼ぶ。

```
move.l    (a0), d0
```

の場合は“a0でポイントされるアドレス（=a0の値）”が実効アドレスとなる。

つまり，“データを指定するのに使われるアドレス”のことを実効アドレスという。

68000にはこの実効アドレス自体を対象とする命令として，leaとpeaが用意されている。

lea（Load Effective Address）は実効アドレスをアドレスレジスタに入れるための専用の命令で，アドレスを扱う都合上，サイズは常にロングワードになる。

```
lea.l    LABEL, a0
```

は“LABELで指定されるアドレスそのもの”をa0レジスタに代入する。動作としては，

```
movea.l #LABEL, a0
```

とまったく変わらない。

また，

```
lea.l     (a1), a0
```

は“a1で指定されるアドレスにあるデータのアドレス（結局a1の値）”をa0レジスタに入れる。つまり，

```
movea.l a1, a0
```

と同じ動作だ。

メモリの読み書きをするとき，CPUはアドレスバスと呼ばれる信号線でアドレスを指定し，その後データバスを介してデータのやり取りをすることになるのだが，leaは“本当はアドレスバスに乗るはずだった値を無理やりデータバスに乗せてデータにしてしまい”アドレスレジスタに入れる命令といえる。

pea（Push Effective Address）は，実効アドレスをスタックでプッシュする命令で，デスティネーションが常に“-(sp)”だということがわかっているから省略して，

```
pea.l   LABEL
pea.l   (a1)
```

のように記述する。それぞれの動作は，

```
move.l  #LABEL, -(sp)
move.l  a1, -(sp)
```

と同じだ。

さて，ここまでの説明ではlea，peaの有用性が見えないと思う(moveで代用できるのだから)。まだ説明していないアドレッシングモードと組み合わせると，カッコよかったり，速かったり，コードが短かったりする場合があるのだが，いまのところはそういうメリットもない。

現時点では“対象がアドレスであることを明確にする意味で使う”ということにしておこうか。

リスト2　STRCPY.S

```
a)   1: *        文字列の複写
     2:
     3:          .include       doscall.mac
     4: *
     5:          .text
     6:          .even
     7: *
     8: ent:
     9:          lea.l    mysp,sp           *spの初期化
    10:
    11:          lea.l    str1,a0           *複写先へのポインタ
    12:          lea.l    str2,a1           *複写元へのポインタ
    13:          bsr      strcpy            *文字列複写
    14:
    15:          pea.l    str1              *コピーした文字列を
    16:          DOS      _PRINT            *表示してみる
    17:          addq.l   #4,sp             *
    18:
    19:          DOS      _EXIT             *終了
    20: *
    21: *文字列複写サブルーチン
    22: strcpy:
    23:          move.b   (a1),d0           *1文字取り出し
    24:          move.b   d0,(a0)           *転送
    25:          addq.l   #1,a0             *転送先ポインタを進める
    26:          addq.l   #1,a1             *転送元ポインタを進める
    27:          cmpi.b   #0,d0             *転送したのは終了コードか?
    28:          bne      strcpy            *そうでなければ繰り返す
    29:          rts
    30: *
    31: str2:    .dc.b    '1234ABCD',0      *複写元
    32: str1:    .ds.b    256               *複写先
    33: *
    34:          .stack
    35:          .even
    36: *
    37: mystack:
    38:          .ds.l    256               *スタック領域
    39: mysp:
    40:          .end

b)  21: *文字列複写サブルーチン
    22: strcpy:
    23:          move.b   (a1)+,d0          *1文字取り出し
    24:          move.b   d0,(a0)+          *転送
    25:          tst.b    d0                *転送したのは終了コードか?
    26:          bne      strcpy            *そうでなければ繰り返す
    27:          rts
    28: *
    29: str2:    ~

c)  21: *文字列複写サブルーチン
    22: strcpy:
    23:          move.b   (a1)+,d0          *1文字取り出し
    24:          move.b   d0,(a0)+          *転送
    25:          bne      strcpy            *終了コードまで繰り返す
    26:          rts
    27: *
    28: str2:    ~

d)  21: *文字列複写サブルーチン
    22: strcpy:
    23:          move.b   (a1)+,(a0)+       *1文字転送
    24:          bne      strcpy            *終了コードまで繰り返す
    25:          rts
    26: *
    27: str2:    ~
```

　　lea.l　　str2, a1

により，a0, a1はそれぞれの領域へのポインタとして使えるようになる（a0, a1は間接的にstr1, str2以下のメモリ領域を表す代名詞のようなものと考える）。こうお膳立てを整えてから文字列を複写するサブルーチンを呼び出す。これは関数へパラメータを渡すようなものだ。

実際のサブルーチン側の中身は次のようになる。

1)　a0の指すアドレスから1バイト取り出し，a1の指すアドレスに転送する

2)　次の文字に備えてa0, a1ともに1を足す

3)　“いま転送した1バイトデータ”が00Hでなければ文字列はまだ続いていることになるので1)に戻る

4)　00Hであれば転送は終了した

ここで，文字列終了コードの00Hも転送に含めなければならないことに注意してもらいたい。00Hを転送しないとどうなるかを考えれば，その理由は明らかだ（よね？）。

この話をそのままプログラムにすると，リスト2-aのようになる。ラベルstrcpy以下が文字列をコピーするサブルーチンだ。

転送元文字列はラベルstr2以下に.dc.bで，転送先領域はstr1以下に.ds.bでそれぞれ用意している。転送先は転送される文字列が十分格納できるだけの大きさを確保しておかなければならない。ここではゆとりを持って256バイト取ってある。

さて，“ポストインクリメント・アドレスレジスタ間接形式”を利用すれば，22～29行のあたりはリスト2-bにまで簡略化できる。どさくさに新しい命令tstを使っているが，(tstはTeSTの略)，これは，

　　cmpi. x　#0, ～

とまったく同じ働きをする（つまり，0との比較専用の）命令だ。データが0かどうか，また正か負かを調べるのに使う。

で，せっかく出てきたtst命令だが，move 命令は，“データを転送し，同時に0と比較してフラグに反映する”ので（さらっと書いたが，これは重要），いまの場合はtstが省略できて，リスト2-cになる。

さらに，moveでフラグが変化するのなら，いちいち“データをレジスタに取り出し，転送して，0と比較する”必要もなくなり，最終的に文字列転送処理はリスト2-dの形になる。ここまで簡単になってしまうと，もうカッコイイとしかいいようがない。

文字列の連結

文字列の複写ができれば連結も簡単だ。上の例同様a0に転送先，a1に転送元文字列の先頭アドレスが入っているとすると，

1)　a0が転送先文字列の最後を指すようにポインタを進める。これはa0がポイントするアドレスから順に00Hを探すことで行う

2)　あとは文字列複写と同等の処理を繰り返す

つまり，転送先文字列の末尾の00Hを上書きする位置以降に文字列を転送すれば，文字列の連結が行える。

プログラムにすると，リスト3のようになる。22～25行が終了コードを探す処理だ。ポストインクリメントしている関係で，00Hが見つかった時点でのa0レジスタは“00Hの次のアドレス”を指しているから，つじつま合わせに25行でa0から1を引いている。

文字列の比較

続いて，指定された2つの文字列が等しいか等しくないかを判断するサブルーチンを作ってみよう。理屈は単純で，2つの文字列の先頭から1バイトずつ比較して，最後まで一致したら2つの文字列は等しいし，途中で不一致の箇所が見つかれば等しくない，ということだ。

リスト4はa0, a1で指定される2つの文字列を比

cmpm命令（末尾の“m”はMemoryの略）は2つのアドレスレジスタでポイントされるメモリ領域を比較する命令で，アドレッシングモードは必ずポストインクリメント・アドレスレジスタ間接形式になる。つまり，

　cmpm. X　(an)＋, (an)＋

の形でだけ使われる。なお，例によってAS. Xでは，

　cmp. b　(a0)＋, (a1)＋

と書けば

　cmpm. b　(a0)＋, (a1)＋

と解釈してくれる。

較するサブルーチンの例だ。2つの文字列の長さが等しいとは限らないので，28行で比較文字列（どちらでも同じことだが便宜上a1のほう）の最後まで比較がすんだかどうかを調べている。まだ文字列が残っていれば30行のcmpm命令で比較を行い，1文字でも不一致ならすぐにループを抜ける。

被比較文字列のほう（a0でポイントされるほう）が先に終わってしまったときのことが心配かもしれないが，その場合は被比較文字列の終了コード 00_H と，比較文字列中の1文字（00_Hではない）が比較されることになり，ちゃんと不一致が検出される。

また，28行で被比較文字列が終わっていることがわかった場合は単純に不一致にしてしまわないで，“もしかすると比較文字列も終わっているかもしれない”ので34行でラストチャンスを与える。

最終的にこのサブルーチンは文字列が一致したらZ=1，不一致ならZ=0でリターンする。あとはサブルーチンから戻ったあと，beq，bneで処理を振り分ければよい。リスト4では文字列が一致したかどうかに応じてそれなりのメッセージを表示するようにしてある。

文字列を適当に変えて試してみてほしい。デバッガで動作を確認するのも面白いだろう。

また，BASICの文字列比較やCのstrcmpでは“文字列の大小比較”が行えるが，いま作ったサブルーチンは，期せずしてこれにも対応している。リターン時のccrのZビットとCビットの組み合わせで，文字列の大小がわかるのだ。

文字列の長さを得る

文字列連結のときに，文字列の先頭から順に 00_H を探す処理が出てきた。文字列の長さを知りたければ，その“00_Hを探す処理”の過程で何バイト飛ばしたかを数えればよい。

リスト5は文字列の長さを数え，結果が何バイトかを表示するプログラムの例だ。

長さを数えるサブルーチンstrlenは，d0.wをカウンタにして文字列の先頭から 00_H までのバイト数を数えている（00_H自体は数えない）。結果はそのままd0.wに返す。カウンタがワードなので，文字列は65535文字以下でなければならないが，実用上の問題はないだろう。

20行でカウンタに使うd0.wを−1で初期化する。0ではなく−1で初期化するのは，22～25行で終了コードをチェックするのに先立ってカウントアップするためだ。

この順序を逆にし，

```
tst.b    (a0)+     *テスト
addq.w   #1, d0    *カウントアップ
bne      ～
```

とやってしまうと，せっかくtstで変化させたフラグがaddqでさらに変化してしまい，正しい結果が得られない。

このサブルーチンよりも28行以下の“d0.wを10進左詰めで表示するサブルーチン”のほうが複雑なので，ちょっと脱線して次節で解説を加える。

リスト3　STRCAT.S

```
 1: *       文字列の連結
 2:
 3:         .include        doscall.mac
 4: *
 5:         .text
 6:         .even
 7: *
 8: ent:
 9:         lea.l   mysp,sp             *spの初期化
10:
11:         lea.l   str1,a0             *連結先へのポインタ
12:         lea.l   str2,a1             *連結元へのポインタ
13:         bsr     strcat              *文字列連結
14:
15:         pea.l   str1                *連結した文字列を
16:         DOS     _PRINT              *表示してみる
17:         addq.l  #4,sp               *
18:
19:         DOS     _EXIT               *終了
20: *
21: *文字列連結サブルーチン
22: strcat:
23:         tst.b   (a0)+               *(a0)は0か？
24:         bne     strcat              *そうでなければ繰り返す
25:         subq.l  #1,a0               *行きすぎたから1つ戻る
26: strcpy:
27:         move.b  (a1)+,(a0)+         *1文字転送
28:         bne     strcpy              *終了コードまで繰り返す
29:         rts
30: *
31: str2:   .dc.b   '1234ABCD',0        *連結元
32: str1:   .dc.b   '5678EFGH',0        *連結先
33:         .ds.b   256                 *ゆとり
34: *
35:         .stack
36:         .even
37: *
38: mystack:
39:         .ds.l   256                 *スタック領域
40: mysp:
41:         .end
```

リスト4　STRCMP.S

```
 1: *       文字列の比較
 2:
 3:         .include        doscall.mac
 4: *
 5:         .text
 6:         .even
 7: *
 8: ent:
 9:         lea.l   mysp,sp             *spの初期化
10:
11:         lea.l   str1,a0             *非比較文字列へのポインタ
12:         lea.l   str2,a1             *比較文字列へのポインタ
13:         bsr     strcmp              *文字列比較
14:
15:         beq     match               *一致したか？
16:
17: not:    pea.l   notmes              *一致しなかった
18:         bra     match0
19:
20: match:  pea.l   matmes              *一致した
21: match0: DOS     _PRINT              *それなりのメッセージを
22:         addq.l  #4,sp               *       表示
23:
24:         DOS     _EXIT               *終了
25: *
26: *文字列比較サブルーチン
27: strcmp:
28:         tst.b   (a1)                *比較文字列は終わりか？
29:         beq     strcmp0             *そうであればループを抜ける
30:         cmpm.b  (a1)+,(a0)+         *1文字比較
31:         beq     strcmp              *一致している間繰り返す
32:         rts                         *一致しなかった
33: strcmp0:
34:         cmpm.b  (a1)+,(a0)+         *ラストチャンス
35:         rts
36: *
37: str1:   .dc.b   '1234ABCD',0        *非比較文字列
38: str2:   .dc.b   '1234ABCD',0        *比較文字列
39: *
40: matmes: .dc.b   '一致しました',$0d,$0a,0
41: notmes: .dc.b   '一致しません',$0d,$0a,0
42: *
43:         .stack
44:         .even
45: *
46: mystack:
47:         .ds.l   256                 *スタック領域
48: mysp:
49:         .end
```

リスト5　STRLEN.S

```
 1: *           文字列の長さを数え表示する
 2:
 3:             .include        doscall.mac
 4: *
 5:             .text
 6:             .even
 7: *
 8: ent:
 9:             lea.l   mysp,sp         *spの初期化
10:
11:             lea.l   strl,a0         *文字列へのポインタ
12:             bsr     strlen          *文字列の長さを数える
13:
14:             bsr     prtdec          *結果を10進表示
15:                                     *改行はしていない
16:             DOS     _EXIT           *終了
17: *
18: *文字列の長さを数えるサブルーチン
19: strlen:
20:             moveq.l #-1,d0          *カウンタの初期化
21: strlen0:
22:             addq.w  #1,d0           *カウント
23:             tst.b   (a0)+           *終了コードか？
24:             bne     strlen0         *そうでなければ繰り返す
25:             rts
26: *
27: *D0.Wを10進左詰めで表示するサブルーチン
28: prtdec:
29:             movem.l d0/a0,-(sp)     *d0,a0をスタックに待避
30:
31:             andi.l  #$0000ffff,d0   *上位ワードをクリア
32:             lea.l   bufend,a0       *ポインタ初期化
33: prtdec0:
34:             divu.w  #10,d0          *d0.lを10で割る
35:                                     *       上位ワード = 余り
36:                                     *       下位ワード = 商
37:             swap.w  d0              *上位ワードと下位ワードを交換
38:                                     *       上位ワード = 商
39:                                     *       下位ワード = 余り
40:             addi.w  #'0',d0         *0～9 → '0'～'9'
41:             move.b  d0,-(a0)        *1桁格納
42:             clr.w   d0              *次の除算に備える
43:                                     *       上位ワード = さっきの商
44:                                     *       下位ワード = 0
45:             swap.w  d0              *上位ワードと下位ワードを交換
46:                                     *       上位ワード = 0
47:                                     *       下位ワード = さっきの商
48:             bne     prtdec0
49:
50:             move.l  a0,-(sp)
51:             DOS     _PRINT
52:             addq.l  #4,sp
53:
54:             movem.l (sp)+,d0/a0
55:             rts
56: *
57: strl:       .dc.b   '1234567890ABCDEFGHIJK',0       *テスト文字列
58: *
59: buff:       .ds.b   5               *10進文字列格納領域
60: bufend:     .dc.b   0               *文字列の終了コード
61: *
62:             .stack
63:             .even
64: *
65: mystack:
66:             .ds.l   256             *スタック領域
67: mysp:
68:             .end
```

movem の末尾の m は Multiple registersの略で，“たくさんのレジスタ”程度の意味。movem では複数のレジスタを“/”で区切って指定することになっているが，d0～d2，a2～a4をまとめて待避するような場合には，

movem.l d0-d2/a2-a4,-(sp)

と書くことが許されている。ちなみに，“d0/a0”や“d0-d2/a2-a4”の部分のことを“レジスタリスト”と呼ぶ。

バッファ(buffer)は“データを溜めておく場所”の意味。

10進表示

リスト5のprtdecはd0.wを無符号数と見なして10進左詰めで表示するサブルーチンだ。一応汎用性を狙ったので，結構がっちり作ってある。

話を簡単にするために，d0.wには0～9の10進1桁で表せる範囲の数しか入っていないものとすると，d0.wに“0”のASCIIコードを足して表示すれば10進表示が行える。

また，d0.wが0～99の範囲に収まるのなら，

1)　10で割った商に“0”のASCIIコードを足して表示
2)　1)の余りに“0”のASCIIコードを足して表示

という2段階の処理で10進2桁表示が行われる。

いま，d0.wには0～65535の数が入っている可能性があるわけだから，10進2桁の場合を拡張して，次のような手順で10進5桁での表示が行えるだろう。

1)　10000で割った商に“0”のASCIIコードを足して表示
2)　1)の余りを1000で割った商に“0”のASCIIコードを足して表示
3)　2)の余りを100で割って表示
4)　3)の余りを10で割って表示
5)　4)の余りを1で割って（結局そのまま）表示

このアルゴリズムでは表示する数の上位桁から順に1桁ずつ取り出しているが，逆に下位桁から取り出す手順もある。

1)　10で割った余りに“0”のASCIIコードを足し，結果をどこかにしまっておく
2)　1)の商が0でなければ1)に戻る
3)　1)～2)により表示すべき文字が1の位から順に求まるので，最後にこれを逆順に表示する

以上2つの方法を比べると，前者は上位桁から処理するからその場その場でどんどん表示できるが，10000で割って，1000で割って……というあたりが冗長な感じがする。対して後者は，逆順になるため一度どこかのメモリに文字を溜めておく必要がある半面，単純なループで処理することができる。

どちらも一長一短があるわけだが，リスト5では後者の方法を採った。

さて，アルゴリズムはこれでいいとして，それをプログラムで実現するにあたってはマシン語で割り算する方法を知らなければならない。幸いなことに68000には除算専用の命令divuがあるので，これを使わせてもらおう（「除算命令」のコラムを参照してほしい）。

ではリスト5のprtdecサブルーチンを頭から順に見てもらいたい。最初にいきなり，

movem.l　d0/a0，−(sp)

という変な命令がある。書式からだいたい見当はつくと思うが，これはレジスタをまとめてスタックにプッシュするのに使う命令だ。ここではd0，a0の2つをスタックに待避している。また，待避したレジスタの値はサブルーチンの最後で，

movem.l　(sp)＋，d0/a0

により，やはりまとめて復帰している。

このレジスタの待避・復帰は汎用性のあるサブルーチンを作るときの常套手段だ。つまり，サブルーチンの中で使うレジスタをメインルーチンで使っている場合に備えて，値を保存しておくわけだ。こうしておけば，メインルーチンで使っているレジスタを誤って壊してしまう心配がなくなり，サブルーチンの独立性が高くなる。ほかのプログラムに組み込むときにも，ただそのまま持っていけばいい。

続いてサブルーチンの初期化部分。まず，

andi.l　#$0000ffff，d0

により，d0.lの下位ワードはそのままで上位ワードを0にしておく。これはあとでdivu命令で割り算をするのに必要な処理だ。

さらに，

lea.l　bufend，a0

により，数値を“数字を表す文字”に変換した結果

を格納するバッファ領域へのポインタを初期化する。いま表示するのは0〜65535までの5桁の数なので，バッファも5バイト用意しておけばよい。

ポインタがバッファの先頭ではなく最後を指すようにしているのは，さっきいった“逆順”の関係だ。表示のときにひっくり返すのではなく，最初から逆に格納しておこうというわけ。これにより，変換結果を最後にまとめて“文字列”として表示することができる。バッファの直後に.dc .bで文字列の終了コード00Hをあらかじめ書き込んであるのもこのためだ。うーん，深慮遠謀。

33行以下が数字を1桁ずつ取り出すメインループ。

```
divu.w   #10, d0
```

により，d0を10で割り，d0の下位ワードに商，上位ワードに余りを求める。

とりあえず必要なのは，上位ワードに入っている余りのほうなので，

```
swap.w   d0
```

により，d0の上位ワードと下位ワードを交換する。この段階でd0.wには10で割った余りが入っているから，“0”のASCIIコードを足し，a0の指す領域へ格納する。a0に対してプリデクリメント・アドレスレジスタ間接アドレッシングを適用しているのがポイントだ。

ここで次のループに備え，ふたたび，

```
swap.w   d0
```

で上位ワードと下位ワードを交換するのだが，その前に，

```
clr.w    d0
```

でd0の下位ワード（すぐひっくり返すから結果として上位ワード）を0でクリアしておく。これは次の除算に備えるためで，冒頭の

```
andi.l   #$0000ffff, d0
```

と同じような役割を果たしている。

clr（当然CLeaRの略）はデスティネーションを0クリアする命令で，

```
clr.X    〜
```

は，

```
move.X   #0, 〜
```

と同じ動作をする。0を代入する専用命令だから，一般にmoveを使うよりも速くて短いコードになる。唯一の例外は“データレジスタの32ビットを0にする”場合で，このときは，

```
clr.l    d0
```

よりも

```
moveq.l  #0, d0
```

のほうが速い。また，clrはアドレスレジスタに対しては使えないが，AS.Xでは，

（次のページに続く）

クイック・イミディエイト・アドレッシング

move, add, subのバリエーションにmoveq, addq, subqというのがあって，クイック・イミディエイト・アドレッシングはこれら3命令だけで使えるアドレッシングモードだ（いうまでもなくqはQuickの頭文字）。AS.Xでは“a付き”，“i付き”の命令同様，単にmove, add, subと書いておけば，使えるところでは適切に“q付き”に置き換えてくれるから，無理に覚える必要はないのだが，無視するわけにもいかないので一気に解説しておく。

イミディエイトというからには，イミディエイトデータ（即値）を扱うアドレッシングモードであり，それぞれの書式は，

```
moveq.l   #1, d0
addq.l    #1, d0
subq.l    #1, a0
```

のようになる。ふつうのイミディエイトアドレッシングとの違いは“コードが短く実行速度が速い”という1点に尽きる。たとえば，

```
move.l    #1, d0
```

が

```
2F00  0000  0001
```

の3ワードのコードに変換されるのに対して，

```
moveq.l   #1, d0
```

は，

```
7000
```

の1ワードですむ。コードが短くなる分メモリから命令コードを読み出す時間も短くなるので，実行速度も3倍になる。

なら何でもかんでもmoveqを使えばいいじゃないかと思った人，残念でした。クイック・イミディエイト形式は“扱える数の範囲などに制限がある”のだ。

そもそも，

```
move.l    #1, d0
```

が3ワードものコードになってしまうのは，1というロングワードのデータもコード化しなければならないからだ。68000の命令コードはワード単位なので，1ワードを命令自体に，2ワード（＝1ロングワード）をデータに当てると，合計3ワードになってしまう。

ちなみに命令自体を表す1ワードのコードはビットフィールドになっていて（数ビット長の“フィールド”をいくつかまとめて1ワードにしてある），2ビットでmoveという命令であることを，2ビットでオペレーションサイズを，また各6ビットずつでソース／デスティネーションオペランドのアドレッシングモードを表すようになっている（詳しくは『アセンブラマニュアル』などを参考に）。

moveqは扱える数の範囲を1バイトの符号つき数で表せる範囲の数（−128〜127）に制限してデータ部分を1バイトに詰め込み，さらにサイズはロングワード固定，デスティネーションオペランドはデータレジスタのみに限定することで，強引に1ワードのコードに抑えているのだ。つまり，データレジスタへロングワードの−128〜127を代入する場合にのみmoveqの恩恵に与れるというわけ。

addq, subqはサイズとデスティネーションオペランドの自由度は下げない代わりに，扱える数の範囲を1〜8に制限して，やはり1ワードのコードにまとめている。1〜8というのは範囲が狭すぎるように見えるかもしれないが，ポインタとして使っているアドレスレジスタを少しずらすとか，（似たようなものだが）DOSコール呼び出し後のスタックポインタを補正するとか，カウンタを増減するなどの用途に使う分には十分だ。

他CPUでは“1を足す命令”，“1を引く命令”が特別に用意されているものだが，68000のaddq, subqはそれらの命令を拡張したものと考えてよい。

なお，moveqに関してはさらに，コラム「符号拡張」も参照してほしい。

符号拡張

先月話した“負の数の表現”を思い出してもらおう。−1は8ビットの2の補数表現で表すと，

11111111B ＝FFH

になるのだった。また，−1を16ビットの2の補数表現で表せば，

1111111111111111B ＝FFFFH

になる。

いま，8ビットの符号つき数を16ビットに変換することを考える。正の数であれば，

8ビット		16ビット
00H(0)	→	0000H(0)
01H(1)	→	0001H(1)
7FH(127)	→	007FH(127)

のように上位に8ビット分の0を補えばいいことはすぐわかるだろう。

ところが負の数は単純に0をつけ足すと，

80H(−128)	→	0080H(128)
FEH(−2)	→	00FEH(254)
FFH(−1)	→	00FFH(255)

というおかしなことになってしまう。この場合は8ビット分の0ではなく1を補って，

80H(−128)	→	FF80H(−128)
FEH(−2)	→	FFFEH(−2)
FFH(−1)	→	FFFFH(−1)

と変換しなければならない。

2の補数表現で表された数の正負は符号ビットが0なら正，1なら負とみなすという約束だったから，上の“正なら8ビットの0を補い，負なら8ビットの1を補う”という規則は，“上位ビットを符号ビットで埋める”とまとめることができる。もちろん，この規則は8ビット→16ビットだけでなく，16ビット→32ビット，8ビット→32ビットなどの場合にもそのまま適用できる。

別項でmoveq命令は内部的にはデータを8ビットで表し，それを32ビットに展開するという話をした。これも符号拡張のなせる技だ。

さて，ここで質問です。

```
moveq.l   #128, d0
```

を実行するとd0レジスタはいくつになるでしょう。また，それはなぜでしょう（答えは載せないよ）。

clr.l　a0
を
suba.l　a0, a0
に置き換えるというアクロバットを見せる。これは少々やりすぎのような気がするが。

swapは交換の結果が0であればZビットを1にするのでbneで処理を振り分ける。交換結果が0でなければ（まだ数字が残っていることになるから）ループし，Z＝1なら交換が完了したことになるからループを抜ける（なおswapに関してはコラム参照のこと）。

変換がすんだ時点でa0は変換結果の文字列の先頭をぴったり指しているから，そのままスタックに積んでDOSコールprintを呼び出せば，表示が完了する。めでたし，めでたし。

フィルタへの第一歩

以上の基本的な文字列操作ができるようになればその応用で，文字列の英大文字→小文字変換やその逆変換（strlwrやstruprに対応），文字列を逆順に並べ換える（strrev），文字列の部分を取り出す（left$, mid$, right$）などの処理も行える。当然これは読者への課題である。

最後に次回へのつなぎとして“キーボードから入力した文字列の英小文字をすべて英大文字に置き換えて表示するという処理をえんえんと続けるプログラム”をリスト6に示す。これは標準入力からの入力に何らかの処理を加えて標準出力に出すフィルタコマンドのシンプルな例になっている。文字列入力待ちのときに行の先頭でCTRL＋Z（CTRLキーを押しながらZのキーを押す）を入力しリターンキーを押すことで終了する。

標準入力からの文字列読み込みにはDOSコール getsを使っている。このDOSコールは，

```
.dc.b    n      ＊入力可能最大文字数
.dc.b    x      ＊入力された文字数
.ds.b    n+1    ＊文字列格納領域
```

の形で用意したメモリ領域の先頭アドレスをスタックに積んでから呼び出す。文字列の格納領域は文字列の終了コードの分も計算に入れ，入力可能最大文字数よりも1バイト余計に用意しておかなければならない。

この領域の先頭アドレスを仮にbuffというラベルで表すと，DOSコールから戻った時点で，

```
d0.l      ←  実際に入力された文字数
buff+1    ←  d0.lと同じ値
buff+2〜  ←  入力された文字列
```

が返される。なお，入力された文字列末尾の改行コ

リスト6　UPPER.S

```
 1: *       英小文字→英大文字変換フィルタの出来そこない
 2:
 3:         .include        doscall.mac
 4: *
 5:         .text
 6:         .even
 7: *
 8: ent:
 9:         lea.l   mysp,sp         *spの初期化
10:
11: loop:   pea     buff            *1行入力
12:         DOS     _GETS           *
13:         addq.l  #4,sp           *
14:
15:         lea     str,a0          *a0=入力文字列先頭
16:         cmpi.b  #$1a,(a0)       *先頭は^Zか?
17:         beq     skip            *そうであれば終了
18:
19:         bsr     strupr          *小文字→大文字変換
20:
21:         pea     str             *変換結果を表示
22:         DOS     _PRINT          *
23:         addq.l  #4,sp           *
24:
25:         pea     crlf            *改行
26:         DOS     _PRINT          *
27:         addq.l  #4,sp           *
28:
29:         bra     loop            *えんえんと繰り返す
30: skip:
31:         DOS     _EXIT           *終了
32: *
33: *英小文字→英大文字変換サブルーチン
34: strupr:
35:         tst.b   (a0)            *文字列の終わりか?
36:         beq     strup1          *そうであれば変換終了
37:
38:         cmpi.b  #'a',(a0)       *英小文字か?
39:         bcs     strup0          *
40:         cmpi.b  #'z'+1,(a0)     *
41:         bcc     strup0          *
42:
43:         subi.b  #$20,(a0)       *小文字なら大文字に変換
44:
45: strup0: addq.l  #1,a0           *ポインタを進める
46:         bra     strupr          *繰り返す
47: strup1: rts                     *サブルーチンからリターン
48: *
49: buff:   .dc.b   255             *入力可能最大文字数
50: cnt:    .dc.b   0               *入力された文字数
51: str:    .ds.b   256             *文字列入力バッファ
52: *
53: crlf:   .dc.b   $0d,$0a,0       *改行コードだけの文字列
54: *
55:         .stack
56:         .even
57: *
58: mystack:
59:         .ds.l   256             *スタック領域
60: mysp:
61:         .end
```

除算命令

68000には割り算を行う命令としてdivuとdivsの2つが用意されている。それぞれDIVide as Unsigned, DIVide as Signedの略で，無符号数と符号付き数の割り算を行う命令だ。符号付き数の除算は滅多に使われないので，まずはdivuのほうだけ覚えておこう（使い方は変わらないけど）。

divuは

```
divu.w  #10, d0
divu.w  d1, d0
```

のようにして使う。サイズはワード固定だ。

デスティネーションは常にデータレジスタであり，32ビットすべてが演算に使用される。また，ソースオペランドは16ビットだけが演算に使われる。演算結果はデスティネーションに指定されたデータレジスタに格納されるが，この格納され方がちょっと変わっていて，下位ワードに商が，上位ワードに余りが返されることになっている。

d0.lが256（＝00000100H）のとき

```
divu.w  #10, d0
```

を実行すると，商が25（＝0019H），余りが6（＝0006H）だから，結果のd0.lは

d0.l＝00060019H

になる。この後

```
move.w  d0, d1
```

とやれば商がd1.wに取り出され，別項で説明するswap命令を使って

```
swap.w  d0
```

によりd0.lの上位ワードと下位ワードを交換してから

```
move.w  d0, d2
```

で今度は余りがd2.wに取り出される。

商か余りが16ビットで収まらなかったときには，演算が失敗した印にccrのVビットを立て（1にして），デスティネーションは変化しない（＝除算は実行されない）。このためdivuは32ビット÷16ビットとはいうものの，実質的には16ビット÷16ビットの演算程度にしか利用できないと考えてよいだろう。

また，0で割ろうとしたときは68000がそれを検出し，X68000の場合は画面中央に“0で除算しました”というエラーメッセージを出してプログラムの実行を中止する。中止されて困るのであれば，divu実行前にソースが0でないことを調べておかなければならないということだ。

ードは自動的に00Hに置き換えられて格納されることになっている。

文字列の小文字→大文字変換はサブルーチンstruprで行っている。a0でポイントされるメモリ領域を順に1バイト取り出して英小文字かどうかを調べ，そうであれば大文字に変換してから格納し直す。

英小文字かどうかの判断は現在注目している1文字のASCIIコードが“a”～“z”に収まっているかどうかで調べている（38～41行)。

さて，このプログラムには手抜きがいっぱいある。まず，全角文字の入力が考慮されていない。全角文字列を適当に打ち込むと，たまに変な文字に化けることがあると思う。

また，標準入力をリダイレクトしたときの動作にも問題がある（あるなんてもんじゃない)。試しに，

A>upper <upper.s

とやってみると，DOSコールprintには本来あるはずのブレイクチェックが効かないことがわかる。当然，^Sによる一時停止も働かない。

しかも，TABコードが“^I”という2文字になっているし，最後まで表示した時点で黙りこくってしまう（改行コードが入力されるのを待ち続けているのだ)。止めるには本体上面にあるINTERRUPTスイッチ（おっと，PROでは本体前面か）のお世話になるしかない。

この辺のことを教訓にして，次回は正しいフィルタの作り方を紹介する予定でいる。

●

今月は新しい命令もビシバシ出てくるわ，分量も結構あるわで，一見盛りだくさんの内容となった（派手さには欠けるけど)。消化不良を起こした読者のために胃薬でも用意してあげたい気分だ。

しかし，順を追って見てもらうと，部分部分ではたいしたことをしているわけではないし，覚えなければならないことなんてほとんどない（いっぱいあると思った人は現在の学校教育に毒されている！)。もしかするとそろそろ専門用語でつまずく人が出るかもしれないが，一度はどこかで解説してあるはずだから，前回までの記事を読み直せば“僕のせいじゃない”ことがわかってもらえるだろう。

と，強引に締めくくって，次回へとつづく。なお質問などがあれば遠慮なくどうぞ。

swap命令

swapはデータレジスタの上位ワードと下位ワードを交換する命令だ。交換はワード単位なので，オペレーションサイズもワード固定になっている。結局書式は次のような形になる。

```
swap.w d0
```

念のため例を挙げておくと，d0.lが12345678Hのとき，

```
swap.w d0
```

を実行するとd0.l=56781234Hになり，再度

```
swap.w d0
```

を実行すれば，d0.l=12345678Hとなって元に戻る。

今月の余談

ビープ音，ベル，某マシンではブザー。BASICのエラーのときなんかにピーっと鳴るあの音だ。X68000では大胆にもビープ音をAD PCMで鳴らし，PCMデータさえ用意すれば好きな音でビープ音を鳴らすことができるわけで，X68000を手に入れた人が1度はビープ音を変えて遊んでみるとまでいわれている。「ジュワッ」とか「OUCH！」とか「やめて，お兄さん……」とか，ね。

あ，いや，そういう話じゃなくって。

思い出したときに話しておかないと忘れそうなので，ここで，ビープ音を鳴らす方法について話しておく。

調べてみた人にはわかると思うが，『プログラマーズマニュアル』のどこを見てもビープ音を鳴らすDOSコールはみつからない。もっと探すとIOCSコールにAD PCMを鳴らす機能があるのだが，これは一般的なPCMの鳴らし方であって，CONFIG.SYSで設定したビープ音を鳴らすものではない。

じゃ，どうやるかというと，コントロールコードを使って鳴らすことになっていたりする。0DH，0AHというコントロールコードを使って改行を行ったように，07Hというコードを“画面”に出力すればビープ音が鳴る。

標準出力がリダイレクトされていない前提であれば，

```
move.w  #$07,-(sp)
DOS     _PUTCHAR
addq.l  #2,sp
```

によってベルを鳴らすことができる。

しかし，標準出力がファイルにリダイレクトされていたりすると，ファイルに07Hというコードが埋め込まれるだけで音は出ない（試してみること)。あくまで“画面”にコントロールコードを送らなければならないのだ。

画面に直接文字を出力するには，標準出力の代わりに“標準エラー出力”を使う（コマンドモードからリダイレクトする方法がないので通常は画面に直接対応している)，IOCSコールの文字表示ファンクションを利用する（OSを通さないからOSの機能であるリダイレクションは働かない）などの方法もあるが，DOSコールにも画面への直接出力を行うものがちゃんとある。

以前に1度使ったDOSコールconctrlはいくつかの機能を兼ねていて，その中のひとつに画面への1文字直接出力が用意されている。これを利用すると，ビープ音を鳴らす処理は，

```
move.w  #$07,-(sp)   *ベルのコード
clr.w   -(sp)        *モード0
DOS     _CONCTRL     *1文字出力
addq.l  #4,sp        *sp補正
```

というように書ける。

話はぐっと軽くなる。僕は最近ある仮説を立てた。それは「プログラマのセンスはビープ音の使い方に現れる」というものだ。

センスのよいプログラマはビープ音にまで気を使って適切なところで適切に鳴らす。ときにはあんまりタイミングよく鳴るもんだから「鳴らしてくれてありがとう」という気にさえなる（実話である。が，どのプログラムだったかは忘れてしまった)。

対して“そうでないプログラマ”は，どういうわけかやみくもにベルを鳴らしたがり，エラーが発生したときや無効なキー入力があったときはもちろん，ひどいのになるとプログラム起動時にピー，処理の合間にピー，実行終了間際にピー，とビープ音を鳴らしまくる。

気持ちはわからないでもない。メッセージを表示するだけではなんとなく寂しいような気がしてアクセントをつけたくなるのだろう。が，あんまりピーピー（もしくは「ジュワッジュワッ」とか「やめてお兄さん，やめてお兄さん」)鳴らされると，だんだんいらいらしてきてしまう。

では回数が少なければそれでいいかといえばそうでもない。“最悪のタイミングで鳴らすわずか1回のビープ音”がすべてをブチ壊しにすることだってあるのだ。

「たかがビープ音ごときにそんなに目くじら立てることはないんじゃない？」という意見もあるだろうが，無神経にビープ音が鳴るようなプログラムはきっとほかの部分も無神経に作ってあるだろうし，ひょっとするとバグがいくつも潜んでいるかもしれないじゃないか（説得力ない？)。

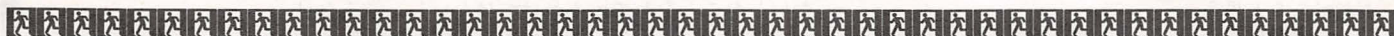
さて，時を同じくして，僕はまた別の仮説を立てた。いわく「プログラマのセンスは画面の色使いに表れるの法則」である。

起動メッセージが意味もなく青かったりするのはまだ可愛いが，エラーメッセージが“黄色のリバース＋強調”だったりすると，作ったヤツの頭を思いきりド突きたくなる。

ま，そういう話。

清く正しくズリズリと(その1)

満開製作所 祝 一平
Iwai Ippei

さて，このC調言語もようやく本格的といおうか，前向きの姿勢というべきか，数回の連載形式で，なんと，エディタを作ってしまうんだそーです。それと，本文とはまったく関係ありませんが，今月は最後にC compiler PRO-68K関連のバグ情報も掲載してありますので，そちらも併せてご覧になってください。

今月から3～5回ぐらいの予定でエディタを作るのである。

そのエディタであるが，基本的な骨組みはできており，普通に使う分に関しては問題ないぐらいのところまできている（検索と置換はまだだけど)。あとはちまちまとデバッグしながら拡張をしていけばよいわけだ。というわけで，まとまりのよいところで区切りつつ，ネチネチと解説していく予定である。

ちなみにエディタの名前は「サンダーワード0号」である。

基本仕様

○Emacs系である

エディタといってもいろいろあるわけだが，今回作るサンダーワード0号は，誰がなんといってもEmacs系である。すなわち，カーソル移動などのコントロールはCTRL-Bで左，CTRL-Fで右，CTRL-Pで上，CTRL-Nで下などという，

黄金律に従う。

文句をいう奴は許さない。どうしてもいやだというのなら，簡単だから自分で勝手に作り変えていただきたい。

○制限がある

それから，残念なことであるが，扱えるテキストの大きさと行数には制限がある。本当の清く正しいエディタとゆーものは，原則としてメモリが許す限りの大きさのテキストを編集できなければならないのだが，今回作ったものはそうなっていない。

この手抜きは，メモリ管理で苦しむのが嫌だったからである。ただし，データアクセスに関しては，オブジェクト指向といえなくもないような形にしてあるので，その気になれば対応は容易であろう(よくある言いのがれ)。というわけで，扱える行数はe.hで定義してあるMaxTextLineで制限されている（結局は配列の大きさである)。また，画面の大きさは横がXwidth，縦がYwidthで決められている。最近ではこの桁×行も動的に変化できるのがはやりであるらしいが，そうなっていない（もちろん頑張って作り直せばできるけど)。ここら辺については，コンパイルし直せば好きなようにできるから勘弁していただきたい。なお，Xwidthの最大は96，Ywidthの最大は32である。編集中のテキストはYwidth－3行分だけ表示される。

○日本語に対応している

当たり前のことであるが，全角/半角が混合した文書に対応している。そして，長い行が右端にぶつかったなら，折れ曲がって，左端から出てくるようになっている。正式な用語かどうかは知らないが，「行はロールする」わけだ（そもそも正式な用語なんてあるのかどうか知らないが)。この点に関しては，ED.Xでは長い行が画面の右からはみ出してしまうので，誰もがウギギになった経験があるだろうから，ちとキバッてみたのである。

で，このように1行が清く正しく折れ曲がっているので，文章の挿入/削除にともない，折れ曲がった行が自動的にズリズリするのである。これは結構重い処理なので，うかつにやるとんでもなく遅いエディタが出来上がることになってしまうが，うかつじゃなく作ったので許してもらえる速度であると思う。

速度を稼ぐ戦略は，よーするに編集の対象になっている部分(＝画面に見えている部分）でつじつまがあっていればよいのであるから，画面外への送り出しに関しては，「なにがどれだけ送り出されたか」だけを憶えておいて，画面スクロールが発生したときなど，どうしても整理しなければならなくなったときに，「どっこいせ」と整理を始めるということになっている。面倒だがこのようにしないと操作性がひどいことになってしまう（究極の手段としては，キー入力のない時間を使ってマルチタスク的にシコシコと整理整頓するという手があるが，そんな複雑なアルゴリズムはちょっと無理である)。

なお，どことなく即戦力(SamuRai)と似ていたりするが，それは気のせいではない。

その前にちょっとエディタについてのクギ刺し

いささか強引であるが，エディタについて語るためにはある程度の資格を求めたい。すなわち，

1）ラインエディタを使用した経験があること（何事も経験であるから，必要がなくとも半日ぐらいいじってみることをお勧めする）

2）vi，Emacs，WordStarなどの，古典的名作エディタにある程度精通しており，少なくとも1つはちゃんと使えること（さもないと世界のレベルをまったく知らないということになる）

3）トロンを称賛したことがないこと（別にトロンを非難しろといっているのではない。私がいいたいのは，実際に使ってみたこともないシステムなのに，「カタログスペックだけで誉めるんじゃない！」ということである。少なくとも，もう少し冷静な態度であるべきである。そういう意味で，トロンというのは，その人がコンピュータについて語る資格があるかどうかの試金石だったりする。もっとも，私みたいに見たこともないうちからケナしているのも異常だけどね）

なんでこんなことを書いたかというと，エディタ(ワープロ)に関して困った議論をする人が結構いるみたいだからなのである。というよりも，コンピューター一般に関してであるが，ちょっと知識を得ただけなのに，みょーな自信を持つ人が，ときどきいるのである。たとえば何本かのワープロを使ってみただけなのに，評論家のような口ぶりになったりなどである。実に困ったことである。私がこの目で見たなかには，「アルファベットは８ビットコードだから８ビットコンピュータでも大丈夫だけど，漢字は16ビットコードだから16ビットコンピュータじゃないとうまく扱えないんだ」などと，いい切った人もいた。カラまれるとイヤなので一生懸命笑うのを我慢したんだけどね。

ちなみに「エディタ」として見るならば，一太郎は完全に二流である。しかしながら日本語ワープロとして見るならば「相対的に一流」である。これはジャストシステムの責任というよりも，ほかのワープロメーカーがだらしないせいであろう。私はこのような考え方を持っているのであるからにして，エディタに関して私と議論してみようなどということは考えないでいただきたい。特に私の前で「頻繁に使う機能をファンクションキーに割り当てるのは間違いではない」などといおうものなら，命はないものと思っていただきたい。

双方向リストなのよ

エディタ(ワープロ)を作る際に真っ先に慎重に考えておかなければならないのが，文書データをどのような形で保持するかである。

データの構造を決めるうえで特に重要なのが，挿入や削除が簡単/迅速に行えるかどうかである。

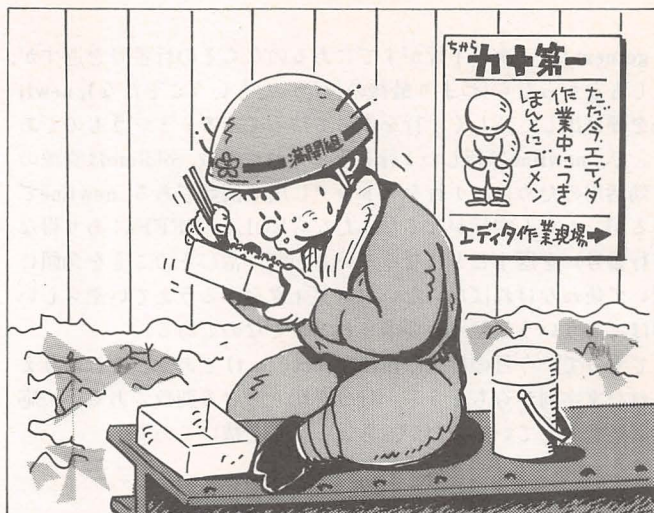
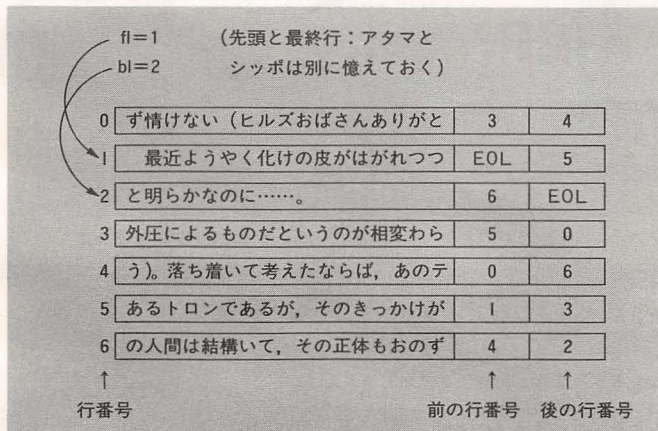
極端な例であるが，文書データをファイルから読み出した形式のまま持っている場合を考えてみよう。そうなると文書の先頭で１文字の挿入/削除があるたびに，文書データ全体(たとえば10Kバイト）をガシガシと転送しなければならないわけである。もしもこんなことをしていたのなら，操作性が地獄を見ることは明らかであろう。

というわけで，エディタの設計は，まずデータ構造から始まるのである。で，このエディタで採用したのは，「固定長文字配列の双方向リスト」である（つまり，e.hで定義してある構造体のLINEだな)。だいたいのところは図１で示すようなものである。ちなみにLISPのリストは片方向リストである。「リスト構造ってなあに」という人は，下図を見て気合で理解してほしい。

さて，データが数珠つなぎになっている場合を考えてみよう。

図１　双方向リストの例

fl＝1　（先頭と最終行：アタマと
bl＝2　シッポは別に憶えておく）

行番号		前の行番号	後の行番号
0	ず情けない（ヒルズおばさんありがと	3	4
1	最近ようやく化けの皮がはがれつつ	EOL	5
2	と明らかなのに……。	6	EOL
3	外圧によるものだというのが相変わら	5	0
4	う)。落ち着いて考えたならば，あのテ	0	6
5	あるトロンであるが，そのきっかけが	1	3
6	の人間は結構いて，その正体もおおず	4	2

そしてひとつの要素にはポインタがひとつだけあるとする（つまり片方向リスト)。このとき，それぞれの要素は「自分の次の要素」は知っているが，自分の前の要素は知らないわけである。そうすると，考えてみるとわかるように，一方向にしかポインタをたどれないのである(注)。このような場合，前の要素を知りたいのなら，リストの先頭からたどり直す必要がある。それはかなりの手間であり，時間的なロスも無視できないことになる。そこで考えるのが自分の次の要素だけではなく，自分の前の要素も知っている「双方向リスト」である。ただしここではポインタではなく，要素の番号(添字)を持っている。このような場合でもリストであることに変わりはない。

行の削除を行う場合は，自分の前後の要素の「自分を指しているポインタ」を書き換えればよい。なお，リストから外した要素は，再利用のために蓄えておく必要がある。プログラムでは未使用(フリー)の要素のリストというのを持っていて，新しいものが必要になるたびにそこから１つずつ外して持ってくることになっている。でもって，行の挿入の場合は逆の操作をすればよいわけだな。

総合的に見て，双方向リストは日本語を扱ってロールするエディタのためのデータ構造としては，かなり適したものだと思われる。それ以上のものも考えられないことはないが，かなり複雑なものにならざるを得ないだろう。

注）世の中には頭のいい人がいて，ポインタ(みたいなもの)をひとつずつしか持っていないのに，双方向にたどれるリスト構造も構成できるということが知られている。具体的にどうやるかは，ここではいわないけど，まさしく「膝ポン」物である。

説明するのである

ではプログラムの説明に入る。なお，ソース中にもたっぷりと日本語で注釈を入れてあるので参考にしていただきたい。

まずは基本的なところからデータの取り扱いをする関数である。前述のように，データは双方向リストに蓄えておくので，そのための関数群がline.cのなかの４つの関数，

```
getnext(l)
newline(m, a)
oldline(l)
collect_free_line( )
```

である。

getnextは，「次の1行がすでにあるのならその行番号を返すが，もしもなかったら(つまり最後の行だったということだな)，newlineを呼び出し，新しく1行を作って持ってくる」というものである。で，newlineは新しい1行を手に入れる関数，oldlineは資源の有効活用のために，1行をストックに戻す関数である。newlineであるが，もしも空行がなくなったならEOL(=FFFF$_H$：あり得ない行番号)を返すことになっているので，常にそのことを念頭に置いて使わなければならない（エディタを作るうえでいやらしいのは，こういう気配りを要求されることなのである)。

でもって，今月の目玉はinsert_cut0(s,t)である。これは与えられた文字列からちょうど1行分を切り分ける関数である。一応禁則処理もしているのだが，ちょっと手を抜いてある。

すなわち，

行頭に「。」や「，」がきてはいけない

という処理はやっているのだが，

行末に「（」や「『」がきてはいけない

という処理はやってなかったりする。

この関数でやっていることは単純で，気をつけなければいけないのは禁則処理とタブの処理ぐらいであるが，この関数こそすべての基本となっている。

getnext( )とよく似たものにnext( )というのがあるが，nextのほうはただ単にリストをたどるだけで，次がない場合は素直にEOLを返すのである。ちょっと見た目には関数みたいになっているが，実はe.hで定義してあるマクロである。

＊　　＊

正直にいうが，実はエディタを作ったのは今回が始めてなのである(けけけけけ)。なぜかというと，一旦作り出せば，結局は腕力の問題だけになることがわかりきっていたからである。だから，何度かてきと一な気分で作りかけたことはあるが，気力の充実がいまいちだったので，「なるほど，こうするとこういう地獄にはまるのだな」という発見をした段階であきらめてほうり出してしまっていたのである。

で，今月ぐらいのプログラムならば，ちょっとCが使える程度の人であれば誰でも作れてしまうレベルであろう。来月もこの調子だと思ったら大正解である。そう，エディタを作るということは，単なる力仕事なのである。要はあちらこちらでからみあった大きなプログラムを理解し，かつ，変数をちゃんと把握し続けることができるかどうかなのである。結局それがやりやすいような工夫をしておけるかどうかも大事になってくる。

さて，エディタを作ろうとするときに覚悟しておかなければならないことが何点かある。

1）削除関係のコマンド入力（たとえばバックスペースなど）がキーバッファに溜まるような事態は極力避けること

2）できるだけ人間を待たせないこと。待たせざるを得ないときは，メッセージを表示すること

3）きれいなプログラムにできるとは金輪際思わないこと

4）サブルーチンの汎用性と，拡張性に最大限の配慮をすること

(エディタに限らずいかなるプログラムでもいえることだが，エディタの場合はこの点が特に重要になってくる)

てなところで，また来月。ばっはは一い。

プログラマーズマニュアルの誤植とSRAMのクリア

C compiler PRO-68K，および福袋Ver.2.0に付属するプログラマーズマニュアルに誤植があります。P.499SRAMアドレス$ED001Cの図ではD07が1になっていますが，これは0の間違いです。このビットを立ててキーボードを初期化するとブレイクキーが入った状態でSRAMが初期化されてしまいます。一部の市販ソフトでこれを行っているため，ある種のソフトをディスクから起動できなくなります(起動時にキー入力要求のあるものの一部)。

昨今，SARMの異常を心配している人も多いようです。$ED0000からの“X68000”の文字を破壊するとSRAMが初期化されることが知られていますが，これだとリザーブ領域は初期化されず，スイッチ類はすべて消えてしまいます。そこでSRAMをクリアするプログラム(ZAP.X)を作りました。このプログラムはオプションによってクリアする範囲を指定でき，キーボードの初期化異常も同時に修正します。(詳しくはソース参照)。終了後，リセットされるので注意しましょう。

また，先ほどのブレイクキーが押された状態で初期化された場合，ソフトによっては起動できなくなります(一部のゲームでブレイクキーを押しながらでないと立ち上がらないものがあった)。このような場合システムディスクを立ち上げて“ZAP/B”と入力すればブレイクキーを押さなくても起動できるようになりますが，それでは面倒なので起動前にチェックするプログラムを作成しました。それがNURSE.Xです。一度コマンドモードからこのプログラムを実行しておけば，あとはSRAMの異常が起こった場合でもそれを修正してからディスクを起動します。また，このプログラムは6月号のDOCTORと同居可能です。

それぞれソースを入力しアセンブルするか，3月号のマシン語入力ツールでダンプリストを打ち込み，990，640バイトにファイルサイズを整えてください。

また，IOCSコール$03“KEY_INIT”で入力パラメータ(D1.B)としてSRAMの図を参照するようになっていますが，サンプルプログラムではD1.Bになにも指定していません。これを参考にするとキーボードLEDの状態が破壊されます。この部分は$02の“B_SFTSNS”で得たキー状態(D0.L)のうちの第8ビットから第14ビットの内容(第15ビットは0に固定)をD1.Bに入れて$03を呼び出すようにしなければなりません。

なお，このIOCSコールは一般ユーザーは使用してはいけません。

（編集部）

リスト1　e.h

```
 1: /* 第1回目↓ */
 2:
 3: #define Xwidth         80      /* 桁数：最高96 */
 4: #define Ywidth         20      /* 行数：最高32 */
 5: #define MaxTextLine    200     /* テキストの最大行数 */
 6: #define MAXLINE        100     /* 作業用の文字列の長さ */
 7: #define Kaigyou        '▼'     /* 改行マーク 81A5H */
 8:
 9: #define txt(X)         (buff[(X)].data)        /* 文字列データ本体 */
10: #define before(X)      (buff[(X)].mae)         /* 前の行番号 */
11: #define next(X)        (buff[(X)].ato)         /* 後の行番号 */
12: #define EOL            (0xffff)                /* 行の終端：ありえない行番号 */
13:
14: typedef struct LINE {
15:         UBYTE data[Xwidth+1];   /* 1 line and EOS */
16:         UWORD mae;              /* 前の行番号 */
17:         UWORD ato;              /* 後ろの行番号 */
18: };
19:
20: /* 第1回目↑ */
```

リスト2　extern.h

```
 1: extern struct LINE buff[];      /* データ本体 */
 2:
 3: extern FILE *fp0;               /* ファイル構造体へのポインタ */
 4: extern UBYTE filename0[];       /* ファイル名 */
 5:
 6: extern int cx,cy;           /* カーソル位置 */
 7: extern int cxk;             /* その行の何文字目にいるか（全角／半角／タブがあるから） */
 8:
 9: extern int fl;              /* first line リストの先頭の行番号 */
10: extern int bl;              /* bottom line リストの最後の行番号 */
```

```
11:
12: extern int hl;            /* home line 画面一番上の行番号 */
13: extern int cl;            /* current line 現在編集中の行番号 */
14:
15: extern int free_line;   /* 未使用の行リストの先頭 */
16: extern int full_flag;   /* 空行がなくなった時用のフラグ */
17:
18: extern UWORD kinsetzen[];         /* 禁則文字のセット */
19:
20: /* 第1回目↑ */
```

リスト3　al.c

```
 1: #include         <stdio.h>
 2: #include         <basic0.h>
 3: #include         <iocslib.h>
 4: #include         <conio.h>
 5: #include         <class.h>
 6: #include         "e.h"
 7: #include         "extern.h"
 8:
 9: main(argc,argv)
10: int argc;
11: UBYTE *argv[];
12: {
13:         init();
14:         if (argc < 2) strcpy(filename0,"DELETE.ME");    /* ファイル名の指定がない */
15:         else strcpy(filename0,argv[1]);
16:         fp0 = fopen(filename0,"r+t");                   /* ファイル更新でオープンしてみる */
17:         if (fp0 == NULL) fp0 = fopen(filename0,"w+t");  /* 新しくファイルを作る */
18:         if (fp0 == NULL) error("main:file open");       /* なんか変だぞ */
19:
20:         getfile(fp0);               /* ファイルの読み込み */
21:
22:         cl = cxk = cx = cy = 0; /* カーソル位置などの設定 */
23:         flush(0);                   /* 位置画面表示 */
24:
25:         core_dump();                /* 今月は単にダンプするだけだったりして */
26:
27:         finis();                    /* コンソールを元に戻したりする */
28: }
29:
30: init()
31: {
32:         UWORD i;
33:
34:         locate(0,Ywidth-3);
35:         printf("%.*s",Xwidth,":::::::::::::::::::::::::::::::::::::::::::::::::::::¥
36: :::::::::::::::::::::::::::::::::::::::::::::");          /* コロンをXwidth個表示する */
37:         B_CONSOL(0,0,Xwidth-1,Ywidth-4);          /* コンソールを標準状態にする */
38:
39:         for(i = 0;i < MaxTextLine;i++) {          /* すべてを空行にする */
40:                 *(txt(i)) = '¥0';
41:         }
42:         collect_free_line();    /* 空行を集める：先頭はfree_line */
43: }
44:
45: finis()
46: {
47:         B_CONSOL(0,0,95,30);
48:         locate(0,Ywidth-2);
49:         fcloseall();
50:         exit();
51: }
52:
53: /* ファイルを読み込む */
54: getfile(fp)
55: FILE *fp;
56: {
57:         int wl,wl0,c;
58:         UBYTE l[MAXLINE*2];         /* MAXLINE > Xwidth */
59:         UBYTE s[MAXLINE*2];
60:
61:         bl = wl = newline(EOL,EOL);                 /* 最初に1行分を手に入れる */
62:         under_print("ファイル読み込み中…");      /* 時間の掛かる作業の時はメッセージを出す */
63:
64:         *s = '¥0';                          /* 前からの繰り越しはなし */
65:         while(! feof(fp)) {
66:                 fgets(l,MAXLINE,fp);    /* 適当な長さだけ読み出す */
67:                 c = strlen(l);
68:                 if (l[(c = strlen(l) - 1)] == '¥n') {   /* 最後が改行なら'▼'にする */
69:                         l[c++] = Kaigyou >> 8;
70:                         l[c++] = Kaigyou & 0xff;
71:                         l[c] = '¥0';                    /* 文字列終了も忘れずに */
72:                 }
73:                 strcat(s,l);            /* 前からの繰り越しの後に加える */
74:
75:         /* 注：MAXLINE > Xwidthだから、sの中には必ず1行分以上の文字列が入っている */
76:
77:                 do {
78:                         insert_cut0(s,l);       /* 一行分を切り分ける */
79:                         strcpy(txt(wl),s);      /* 納める */
80:                         wl = getnext(wl0 = wl); /* 次の行番号を取ってくる */
81:                         if (wl == EOL) {        /* もう空行がない！ */
82:                                 bl = wl0;
83:                                 under_blanc();
84:                                 return;
85:                         }
86:                         strcpy(s,l);            /* sに余りを転送 */
87:                 } while(strlen(s) >= Xwidth);   /* sが十分短くなったら新しく読み出して来る */
88:         }
89:         bl = wl;        /* 最後の行番号をblに覚えておく */
90:         under_blanc();
91: }
92:
93: /* sを一行に入り切る長さに切る。それ以降の分をtに入れる。 */
94: /* 一行の長さ（sの長さ）を返す */
95: insert_cut0(s,t)
96: UBYTE *s,*t;
97: {
98:         int qq,qq0;
```

```
 99:         int c,c1,c2;
100:         int w;          /* 表示された場合の桁数を数える */
101:
102:         qq = 0;
103:         w = 0;
104:
105:         while(1) {
106:                 qq0 = qq;
107:                 c2 = 0;
108:                 if ((c = c1 = s[qq++]) == '¥0') {       /* 文字列が終わった */
109:                         *t = '¥0';                      /* 余りは無し */
110:                         return(qq0);
111:                 }
112:                 if (iskanji(c1)) {      /* 漢字の1バイト目か */
113:                         if ((c2 = s[qq++]) == '¥0') error("insert_cut(1):odd code");
114:                         if (iskanji2(c2)) {
115:                                 c = ((c1 << 8) | c2) & 0xffff;
116:                                         /* cはシフトJISコード */
117:                         } else {
118:                                 error("insert_cut(2):odd code");
119:                         }
120:                 }
121: /* c = code, c1 = high, c2 = low */
122: /*
123:         次の行に移行する場合
124: (0)右側にスペースがない
125:         :w >= Xwidth
126: (1)右側に半角1字分しかスペースがない
127:         :w >= Xwidth-1
128: (2)右側に半角2字分のスペースがあるが、次の文字は禁則文字ではない
129:         :(w >= Xwidth-2) && (! kinsoku(x))
130: */
131: /* （や「などの禁則には対応していない */
132:
133:                 if ((w >= Xwidth-1) ||
134:                      (( w >= Xwidth-2) && (! kinsoku(c)))) {     /* 1行分完了 */
135:                         strcpy(t,&s[qq0]);      /* 残りをtに入れる */
136:                         s[qq0] = '¥0';          /* 文字列の終わり */
137:                         return(qq0);
138:                 }
139:                 switch(c) {
140:                 case '¥t':              /* タブの処理は面倒なのだ */
141:                         w = (w+9) & 0xfff8;
142:                         break;
143:                 case Kaigyou:           /* 改行マーク 81A5H */
144:                         strcpy(t,&s[qq]);
145:                         s[qq] = '¥0';   /* 当然、1行分完了 */
146:                         return(qq);
147:                 default:
148:                         if (c2) {       /* 全角 */
149:                                 w += 2;
150:                         } else {        /* 半角 */
151:                                 w++;
152:                         }
153:                         break;
154:                 } /* endswitch */
155:         }
156: }
157:
158: /* 全画面を書き直す */
159: flush(l)
160: UWORD l;        /* 表示を始める行 */
161: {
162:         unsigned int i;
163:
164: /*      insert_cut_all();       */      /* そのうちわかる */
165:         flush0(0,hl = l);       /* hlを変更していることに注意 */
166: }
167:
168: /* 指定行から下を書き直す */
169: flush0(y,l)
170: int y;          /* 書き直し開始位置 */
171: UWORD l;        /* 書き直し開始行：最終行(bl)ではないこと */
172: {
173: /*      flush_finish = 0; */    /* そのうちわかる */
174:         for(;y < Ywidth-3;y++) {
175:                 flush_1line(y,txt(l));  /* 1行表示 */
176:                 l = next(l);
177:                 if (l == EOL) {         /* 最終行に突き当たった */
178:                         B_PRINT("¥x1b[0J");     /* カーソル位置から最後までクリア */
179:                         break;
180:                 }
181: /*
182:                 if (vz_interrupt()) {
183:                         return;
184:                 }
185: */                              /* そのうちわかる */
186:         }
187: /*      flush_finish = 1; */    /* そのうちわかる */
188: }
189:
190: /* 1行だけyの位置に表示する */
191: flush_1line(y,s)
192: int y;
193: UBYTE *s;
194: {
195:         locate(0,y);
196:         B_PRINT("¥x1b[0K");     /* 1行分クリアする */
197:         B_PRINT(s);
198: /*      どひー。B_PRINTとprintfとでは、速度がちぇんちぇん違うがな。      */
199: /*              printf("%s¥x1b[0K",txt(l));*/
200: }
201:
```

リスト4 sub.c

```
1: #include        <stdio.h>
2: #include        <basic0.h>
3: #include        <iocslib.h>
4: #include        <conio.h>
5: #include        <class.h>
```


▶実際，娘に「電子」という名を付けたら，その子が物心ついたとき，親はタダじゃすまねーだろうなと思う僕。 岩田　大（18）大阪府

```
 6: #include         "e.h"
 7: #include         "extern.h"
 8:
 9: /* コントロール行に表示する */
10: under_print(s)
11: UBYTE *s;
12: {
13:         B_CONSOL(0,0,Xwidth-1,Ywidth-2);         /* コンソールを切り直す */
14:         locate(0,Ywidth-2);
15:         B_PRINT("¥x1b[0K");                      /* 1行クリア */
16:         B_PRINT(s);
17: }
18:
19: /* コントロール行を消す */
20: under_blanc()
21: {
22:         under_print("");
23:         B_CONSOL(0,0,Xwidth-1,Ywidth-4);         /* コンソールを標準状態にする */
24: }
25:
26: /* 全角文字 c が禁則文字かどうかをチェックする */
27: /* ただし手抜きをしてあるので、「や（などには対応していない */
28: kinsoku(c)
29: UWORD c;
30: {
31:         int i;
32:         UWORD w;
33:
34:         i = 0;
35:         while(w = kinsetzen[i++]) {
36:                 if (w == c) return(1);
37:         }
38:         return(0);
39: }
40:
41: /* ビープ音を鳴らす */
42: bell()
43: {
44:         putchar('¥007');
45: }
46:
47: error(s)
48: UBYTE *s;
49: {
50:         bell();          /* ビープ音を鳴らす */
51:         under_print(s); /* エラーメッセージを表示する */
52:         key_empty();     /* キーバッファをクリアする */
53:         getch();         /* キー入力を待つ */
54:         under_blanc();
55: }
56:
57: /* キーバッファをクリアする */
58: key_empty()
59: {
60:         KFLUSHIO(0xff);          /* key buff empty */
61: }
62:
63: /* データを無条件にダンプする */
64: /* ファイル名は"core", 場所はカレントディスクのカレントディレクトリ */
65: core_dump()
66: {
67:         UWORD i;
68:         FILE *core_fp,*target_fp;
69:
70:         core_fp = fopen("core","w");
71:         if (core_fp == NULL) error("コアダンプできない！");
72:
73:         for(i=0;i < MaxTextLine;i++) {
74:         /* データ、行番号、前へのポインタ、後へのポインタ */
75:                 fprintf(core_fp,"%-*s %d %d %d¥n",
76:                 Xwidth+3,txt(i),i,before(i),next(i));
77:         }
78:         fclose(core_fp);
79: }
80:
81:
```

リスト5 value.c

```
 1: #include         <stdio.h>
 2: #include         <basic0.h>
 3: #include         <iocslib.h>
 4: #include         <conio.h>
 5: #include         <class.h>
 6: #include         "e.h"
 7:
 8: struct LINE buff[MaxTextLine];  /* データ本体 */
 9:
10: FILE *fp0;                       /* ファイル構造体へのポインタ */
11: UBYTE filename0[MAXLINE];        /* ファイル名 */
12:
13: int cx,cy;       /* カーソル位置 */
14: int cxk;         /* その行の何文字目にいるか（全角／半角／タブがあるから） */
15:
16: UWORD fl;                  /* first line リストの先頭の行番号 */
17: UWORD bl;                  /* bottom line リストの最後の行番号 */
18:
19: UWORD hl;                  /* home line 画面一番上の行番号 */
20: UWORD cl;                  /* current line 現在編集中の行番号 */
21:
22: UWORD free_line;                 /* 未使用の行リストの先頭 */
23: int full_flag = 0;         /* 空行がなくなった時用のフラグ */
24:
25: UWORD kinsetzen[] = {      /* 禁則文字のセット */
26: '、', '。', '？', '！',
27: '」', '』', '）', '｝', '］', '”',
28: 'ぁ', 'ぃ', 'ぅ', 'ぇ', 'ぉ',
29: 'ァ', 'ィ', 'ゥ', 'ェ', 'ォ',
30: 'っ', 'ッ',
31: 'ゃ', 'ゅ', 'ょ',
32: 'ャ', 'ュ', 'ョ',
```

▶アナログジョイスティックのメカの動きは，まさに一級品，ゲームをしなくても動かしているだけで最高！ みんなも買おうね！ 古川 英治（22）青森県

```
 33: '▼',0
 34: };
 35:
 36: /* 第1回目↑ */
 37:
```

リスト6 line.c

```
  1: #include        <stdio.h>
  2: #include        <basic0.h>
  3: #include        <iocslib.h>
  4: #include        <conio.h>
  5: #include        <class.h>
  6: #include        "e.h"
  7: #include        "extern.h"
  8:
  9: /* lの次の行番号を返す。次の行が無かったら新しく作る */
 10: getnext(l)
 11: UWORD l;
 12: {
 13:         UWORD i;
 14:
 15:         if ((i = next(l)) != EOL) return(i);    /* 次があるのならそれを返す */
 16:         if ((i = newline(l,EOL)) == EOL) {
 17:                 full_flag = 1;
 18:                 bell();         /* ぎょー!  空いてる行がない */
 19:         }
 20:         next(l) = i;            /* ちゃんとリンクしておく */
 21:         return(i);
 22: }
 23:
 24: /* 前が m で後が a の行を新しく作り、その行番号を返す */
 25: newline(m,a)
 26: UWORD m,a;
 27: {
 28:         UWORD r;
 29:
 30:         if ((r = free_line) != EOL) {   /* r==EOLなら、もう空いている行は無い */
 31:                 free_line = next(free_line);    /* 次が先頭となる */
 32:                 before(r) = m;
 33:                 next(r) = a;                    /* ちゃんとリンクしておく */
 34:                 if (m != EOL) next(m) = r;      /* 後の前は自分 */
 35:                 if (a != EOL) before(a) = r;    /* 前の後は自分 */
 36:                 *(txt(r)) = '¥0';               /* 念のためデータを空にしておく */
 37:                 if (a == fl) {
 38:                         fl = r;         /* 後がflなら新しい行がflになる */
 39:                 }
 40:                 if (a == hl) {
 41:                         hl = r;         /* 後がhlなら新しい行がhlになる */
 42:                 }
 43:                 if (m == bl) {
 44:                         bl = r;         /* 前がblなら新しい行がblになる */
 45:                 }
 46:         }
 47:         return(r);
 48: }
 49:
 50: /* l行を未使用にする */
 51: oldline(l)
 52: {
 53:         UWORD m,a;
 54:
 55:         m = before(l);
 56:         a = next(l);
 57:         *(txt(l)) = '¥0';               /* 念のためデータを空にしておく */
 58:
 59:         if (l == fl) {          /* 先頭行だったなら自分の後が先頭行となる */
 60:                 fl = a;
 61:         }
 62:         if (l == hl) {          /* hlだったなら自分の後がhlとなる */
 63:                 hl = a;
 64:         }
 65:         if (l == bl) {          /* 最終行だったなら自分の前が最終行となる */
 66:                 bl = m;
 67:         }
 68:
 69:                 /* 自分を抜いてリンクする */
 70:         if (m != EOL) next(m) = a;      /* 前の後を、自分の後にする */
 71:         if (a != EOL) before(a) = m;    /* 後の前を、自分の前にする */
 72:
 73:         before(free_line) = l;
 74:         next(l) = free_line;
 75:         before(l) = EOL;
 76:         free_line = l;          /* 自分を空行の先頭にする */
 77: }
 78:
 79: /* 空行を集める:先頭はfree_lineにする */
 80: /* 若干の問題あり:テキストの中には空行の物もあるので(たとえばbl)、
 81:    初期化以外の時の使用には注意のこと */
 82: /* 集まった空行の数を返す */
 83: collect_free_line()
 84: {
 85:         UWORD i,last;
 86:         int n;
 87:
 88:         free_line = EOL;
 89:
 90:         for(n = i = 0;i < MaxTextLine;i++) {
 91:                 if (*(txt(i)) == '¥0') {                /* 空行ならリストに加える */
 92:                         n++;                            /* カウントを増やす */
 93:                         if (free_line == EOL) {         /* 先頭を決める */
 94:                                 last = free_line = i;
 95:                                 before(free_line) = EOL;
 96:                         } else {
 97:                                 next(last) = i;         /* リストの後につなぐ */
 98:                                 before(i) = last;
 99:                                 last = i;
100:                         }
101:                 }
102:         }
```

```
103:            if (n) {          /* 何かあった */
104:                    next(last) = EOL;
105:            }
106:            return(n);
107: }
108:
```

S-RAMチェックプログラム

リスト7　S-RAM ZAPPERソースリスト

これらのプログラムは「電脳倶楽部8月号」にも収録されています。

```
  1: *********************************
  2: *                               *
  3: *     S-RAM ZAPPER Ver 1.01     *
  4: *                               *
  5: *        1989/06/25 (日)        *
  6: *                               *
  7: *********************************
  8:
  9:         moveq.l #$81,d0                 * B_SUPER
 10:         clr.l   a1
 11:         trap    #15
 12:
 13:         move.l  d0,d7                   * USER STACK POINTER
 14:
 15:         move.b  #$31,$e8e00d            * S-RAM WRITE ENABLE
 16:         addq.l  #1,a2
 17: loop:
 18:         move.b  (a2)+,d0
 19:         tst.b   d0
 20:         beq     all
 21:         cmp.b   #$20,d0
 22:         beq     loop                    * スペースなら読み飛ばし
 23:         cmp.b   #$09,d0
 24:         beq     loop                    * タブなら読み飛ばし
 25:         cmp.b   #'/',d0
 26:         beq     switch                  * / があったらスイッチ
 27:         cmp.b   #'-',d0
 28:         beq     switch                  * - があったらスイッチ
 29:         bra     help                    * 使い方の説明
 30:
 31: switch:
 32:         move.b  (a2),d0                 * スイッチを取り込む
 33:         andi.b  #$df,d0                 * 英大文字にする
 34:         cmp.b   #'B',d0                 * ブレークキービットだけをねかせる
 35:         beq     break_bit
 36:         cmp.b   #'R',d0                 * リザーブエリアを初期化する
 37:         beq     reserve
 38:         cmp.b   #'P',d0                 * プログラムエリアを初期化する
 39:         beq     prog
 40:         cmp.b   #'D',d0                 * SRAMDISK エリアを初期化する
 41:         beq     ramdisk
 42:         cmp.b   #'A',d0                 * リザーブエリアから後ろを全て初期化する
 43:         beq     all
 44:         bra     help
 45:
 46: break_bit:
 47:         andi.b  #$7f,$ed001c            * ブレークキービットをねかせる
 48:
 49:         moveq.l #$03,d0
 50:         moveq.l #0,d1
 51:         trap    #15
 52:
 53:         lea     brk_mes,a1
 54:         moveq.l #$21,d0                 * B_PRINT
 55:         trap    #15
 56:         bra     ret
 57:
 58: reserve:
 59:         andi.b  #$7f,$ed001c            * ブレークキービットをねかせる
 60:
 61:         moveq.l #$03,d0
 62:         moveq.l #0,d1
 63:         trap    #15
 64
 65:         move.w  #$a4,d0
 66:         lea     $ed005b,a0              * リザーブエリア先頭アドレス
 67:         lea     res_mes,a1
 68:         bsr     clr
 69:         bra     ret
 70:
 71: prog:
 72:         andi.b  #$7f,$ed001c            * ブレークキービットをねかせる
 73:
 74:         moveq.l #$03,d0
 75:         moveq.l #0,d1
 76:         trap    #15
 77:
 78:         move.w  #$3eff,d0
 79:         lea     $ed0100,a0              * プログラムエリア先頭アドレス
 80:         lea     pro_mes,a1
 81:         bsr     clr
 82:         bra     ret
 83:
 84: ramdisk:
 85:         andi.b  #$7f,$ed001c            * ブレークキービットをねかせる
 86:
 87:         move.w  #$3bff,d0
 88:         lea     $ed0400,a0              * SRAM DISK エリア先頭アドレス
 89:         lea     ram_mes,a1
 90:         bsr     clr
 91:         bra     ret
 92:
 93: all:
 94:         andi.b  #$7f,$ed001c            * ブレークキービットをねかせる
 95:
 96:         moveq.l #$03,d0
 97:         moveq.l #0,d1
 98:         trap    #15
 99:
100:         move.w  #$3fa4,d0
101:         lea     $ed005b,a0              * リザーブエリア先頭アドレス
102:         lea     all_mes,a1
103:         bsr     clr
104:         bra     ret
105:
106: clr:
107:         clr.b   (a0)+
108:         dbra    d0,clr
109:         moveq.l #$21,d0                 * B_PRINT
110:         trap    #15
111:         rts
112:
113: ret:
114:         clr.b   $e8e00d                 * SRAM WRITE DISABLE
115:
116:         moveq.l #$81,d0                 * B_SUPER
117:         move.l  d7,a1
118:         trap    #15
119:         clr.l   d0
120:         trap    #10
121:
122: help:
123:         moveq.l #$21,d0                 * B_PRINT
124:         lea     help_mes,a1
125:         trap    #15
126:         clr.b   $e8e00d                 * SRAM WRITE DISABLE
127:
128:         moveq.l #$81,d0                 * B_SUPER
129:         move.l  d7,a1
130:         trap    #15
131:         dc.w    $ff00                   * EXIT
132:
133: help_mes:
134:         dc.b    13,10,'SRAM ZAPPER Ver 1.01 By Y.Miyajima',13,10
135:         dc.b    '使用法 ZAP [スイッチ]',13,10
136:         dc.b    '       /B      ブレークキービットだけをねかせる',13,10
137:         dc.b    '       /R      リザーブエリア($ED005B~$ED00FF)を初期化する',13,10
138:         dc.b    '       /P      プログラムエリア($ED0100~$ED3FFF)を初期化する',13,10
139:         dc.b    '       /D      SRAM DISKエリア($ED0400~$ED3FFF)を初期化する',13,10
140:         dc.b    '       /A      リザーブエリア以降($ED005B~$ED3FFF)を初期化する',13,10
141:         dc.b    ' なお、いずれの場合もブレークキービットチェックを行います',13,10,0
142:
143: brk_mes:
144:         dc.b    'ブレークキービットをねかせました',13,10,0
145: res_mes:
146:         dc.b    'リザーブエリア($ED005B~$ED00FF)を初期化しました。',13,10,0
147: pro_mes:
148:         dc.b    'プログラムエリア($ED0100~$ED3FFF)を初期化しました。',13,10,0
149: ram_mes:
150:         dc.b    'SRAM DISK エリア($ED0400~$ED3FFF)を初期化しました。',13,10,0
151: all_mes:
152:         dc.b    'リザーブエリア以降($ED005B~$ED3FFF)を初期化しました。',13,10,0
```

リスト8　BREAK BIT CHECKERソースリスト

```
  1: *********************************
  2: *      BREAK BIT CHECKER        *
  3: *      NURSE Ver 1.00           *
  4: *                               *
  5: *         1989/06/24            *
  6: *        By Y.Miyajima          *
  7: *********************************
  8:
  9:   moveq.l  #$81,d0          * B_SUPER
 10:   clr.l  a1
 11:   trap  #15
 12:   move.l  d0,ssp
 13:
 14:   move.b  #$31,$e8e00d      * SRAM WRITE ENABLE
 15:
 16:   cmpi.l  #$b781420d,$ed0400  * DOCTOR.R が常駐しているか
 17:   beq  with_doctor
 18:
 19:   lea  $ed0100,a2
 20:   bsr  keep
 21:
 22:   lea  only,a0
 23:   move.w  (a0),(a2)
 24:
 25:   moveq.l  #$21,d0        * B_PRINT
 26:   lea  mes1,a1
 27:   trap  #15
 28:
 29:   bra  ret
 30:
 31: with_doctor:
 32:   lea  $ed0900,a2
 33:   bsr  keep
 34:   move.w  #$07fe,$ed0102  * パッチあて
 35:
 36:   lea  with,a0
 37:   move.l  (a0)+,(a2)+
 38:   move.w  (a0),(a2)
 39:
 40:   moveq.l  #$21,d0        * B_PRINT
 41:   lea  mes2,a1
 42:   trap  #15
 43:
 44:   bra  ret
 45:
 46: keep:
 47:   lea  nurse,a0           * nurseの本体の先頭
 48:   lea  endnurse,a1        * nurseの本体の最後
 49:   suba.l  a0,a1
 50:   move.l  a1,d0           * nurseの本体の長さ
 51:   subq.l  #1,d0
 52: loop:
 53:   move.b  (a0)+,(a2)+     * 転送
 54:   dbra  d0,loop
 55:   rts
 56:
 57: ret:
 58:   move.l  #$ed0100,$ed000c  * ROM BOOT ADDRESS
 59:   move.l  #$ed0100,$ed0010  * RAM BOOT ADDRESS
 60:   move.w  #$b000,$ed0018    * BOOT TYPE
 61:   move.b  #$02,$ed002d      * SRAM = PROGRAM MODE
 62:
 63:   clr.b  $e8e00d            * SRAM WRITE DISABLE
 64:
 65:   move.l  ssp,a1
 66:   moveq.l  #$81,d0        * B_SUPER
 67:   trap  #15
 68:
 69:   dc.w  $ff00
 70:
 71: only:
 72:   rts
 73: with:
 74:   jmp  $ed0104
 75:
 76: ssp:
 77:   ds.l  1
 78: mes1:
 79:   dc.b  'BREAK BIT CHECKER を組み込みました。',13,10
 80:   dc.b  '      By Y.Miyajima',13,10,0
 81:
 82: mes2:
 83:   dc.b  'BREAK BIT CHECKER を DOCTOR とともに組み込みました。',13,10
 84:   dc.b  '          By Y.Miyajima',13,10,0
 85:
 86:   .even
 87: nurse:
 88:   bra  main
 89:
 90: ssp2:
 91:   ds.l  1
 92: chkmes:
 93:   dc.b  'ブレークビットおよび、ウイルスのチェックを行います。',13,10,13,10,0
 94: bitmes:
 95:   dc.b  'ブレークビットをねかせました。',13,10,0
 96: wilmes:
 97:   dc.b  'ウイルスに感染したコマンドを使用した可能性があります。',13,10,0
 98:
 99:   .even
100: main:
101:   movem.l  d0-d7/a0-a6,-(sp)
102:
```

▶今回の出張は予想以上に長く，Oh!Xを出張先で買うはめになりました。ところが田舎ゆえ，置いてある店がなく，Oh!X 1冊を探して本屋のハシゴをするはめになりました。まさしく「Oh!Xを探して三千里」でありました。　瀬谷　裕（29）埼玉県

```
103:   moveq.l  #$21,d0       * B_PRINT
104:   lea  chkmes(pc),a1
105:   trap  #15
106:
107:   moveq.l  #$81,d0       * B_SUPER
108:   clr.l  a1
109:   trap  #15
110:   lea  ssp2(pc),a0
111:   move.l  d0,(a0)
112:
113:   move.b  #$31,$e8e00d  * SRAM WRITE ENABLE
114:
115:   bclr.b  #7,$ed001c    * BREAK BIT CHECK
116:   beq  skip
117:
118:   lea  bitmes(pc),a1
119:   moveq.l  #$21,d0       * B_PRINT
120:   trap  #15
121:
122: skip:
123:   moveq.l  #$03,d0       * B_KEYINIT
124:   moveq.l  #$00,d1
125:   trap  #15
126:
127:   tst.b  $ed00ff        * ウイルスチェック
128:   beq  skip2
129:
130:   lea  wilmes(pc),a1
131:   moveq.l  #$21,d0       * B_PRINT
132:   trap  #15
133:
134:   clr.b  $ed00ff
135:
136: skip2:
137:   movem.l  (sp)+,d0-d7/a0-a6
138:
139: endnurse:
```

リスト9 S-RAM ZAPPERダンプリスト

```
0000  48 55 00 00 00 00 00 00  : 9D
0008  00 00 00 00 00 00 03 A0  : A3
0010  00 00 00 00 00 00 00 00  : 00
0018  00 00 00 0C 00 00 00 00  : 0C
0020  00 00 00 00 00 00 00 00  : 00
0028  00 00 00 00 00 00 00 00  : 00
0030  00 00 00 00 00 00 00 00  : 00
0038  00 00 00 00 00 00 00 00  : 00
0040  70 81 93 C9 4E 4F 2E 00  : 18
0048  13 FC 00 31 00 E8 E0 0D  : 15
0050  52 8A 10 1A 4A 00 67 00  : B7
0058  00 C4 B0 3C 00 20 67 F2  : 29
0060  B0 3C 00 09 67 EC B0 3C  : 34
0068  00 2F 67 0A B0 3C 00 2D  : B9
0070  67 04 60 00 00 E6 10 12  : D3
0078  02 00 00 DF B0 3C 00 42  : 0F
-----------------------------------
SUM:  36 8F 1A 4E 5F A1 9F 5C  A4CA

0080  67 1E B0 3C 00 52 67 34  : 5E
0088  B0 3C 00 50 67 50 B0 3C  : DF
0090  00 44 67 6C B0 3C 00 41  : 44
0098  67 00 00 82 60 00 00 BC  : 05
00A0  02 39 00 7F 00 ED 00 1C  : C3
00A8  70 03 72 00 4E 4F 43 F9  : BE
00B0  00 00 02 A0 70 21 4E 4F  : D0
00B8  60 00 00 90 02 39 00 7F  : AA
00C0  00 ED 00 1C 70 03 72 00  : EE
00C8  4E 4F 30 3C 00 A4 41 F9  : E7
00D0  00 ED 00 5B 43 F9 00 00  : 84
00D8  02 C3 61 62 60 6C 02 39  : 8F
00E0  00 7F 00 ED 00 1C 70 03  : FB
00E8  72 00 4E 4F 30 3C 3E FF  : B8
00F0  41 F9 00 ED 01 00 43 F9  : 64
00F8  00 00 02 F8 61 40 60 4A  : 45
-----------------------------------
SUM:  53 3E 6C 5F DC 18 AE C7  D396

0100  02 39 00 7F 00 ED 00 1C  : C3
0108  30 3C 3B FF 41 F9 00 ED  : CD
0110  04 00 43 F9 00 00 03 2F  : 72
0118  61 24 60 2E 02 39 00 7F  : CD
0120  00 ED 00 1C 70 03 72 00  : EE
0128  4E 4F 30 3C 3F A4 41 F9  : 26
0130  00 ED 00 5B 43 F9 00 00  : 84
0138  03 66 61 02 60 0C 42 18  : 92
0140  51 C8 FF FC 70 21 4E 4F  : 42
0148  4E 75 42 39 00 E8 E0 0D  : 13
0150  70 81 22 47 4E 4F 42 80  : B9
0158  4E 4A 70 21 43 F9 00 00  : 65
0160  01 32 4E 4F 42 39 00 E8  : 33
0168  E0 0D 70 81 22 47 4E 4F  : E4
0170  FF 00 0D 0A 53 52 41 4D  : 49
0178  20 5A 41 50 50 45 52 20  : 12
-----------------------------------
SUM:  45 C9 4E 21 9D 33 49 48  A47C

0180  56 65 72 20 31 2E 30 31  : 0D
0188  20 42 79 20 59 2E 4D 69  : 38
0190  79 61 6A 69 6D 61 0D 0A  : 92
0198  8E 67 97 70 96 40 09 5A  : 35
01A0  41 50 20 5B 83 58 83 43  : AD
01A8  83 62 83 60 5D 0D 0A 09  : 45
01B0  2F 42 09 83 75 83 8C 81  : 02
01B8  5B 83 4E 83 4C 81 5B 83  : 5A
01C0  72 83 62 83 67 82 BE 82  : 03
01C8  AF 82 F0 82 CB 82 A9 82  : 1B
01D0  B9 82 E9 0D 0A 09 2F 52  : C5
01D8  09 83 8A 83 55 81 5B 83  : 4D
01E0  75 83 47 83 8A 83 41 28  : 38
01E8  24 45 44 30 30 35 42 81  : 05
01F0  60 24 45 44 30 30 46 46  : F9
01F8  29 82 F0 8F 89 8A FA 89  : C0
-----------------------------------
SUM:  D0 5E 6B F5 32 66 BB 9F  A1EC

0200  BB 82 B7 82 E9 0D 0A 09  : 7F
0208  2F 50 09 83 76 83 8D 83  : 14
0210  4F 83 89 83 80 83 47 83  : AB
0218  8A 83 41 28 24 45 44 30  : 53
0220  31 30 30 81 60 24 45 44  : 1F
0228  33 46 46 46 29 82 F0 8F  : 2F
0230  89 8A FA 89 BB 82 B7 82  : 0C
0238  E9 0D 0A 09 2F 44 09 53  : D8
0240  52 41 4D 20 44 49 53 4B  : 2B
0248  83 47 83 8A 83 41 28 24  : E7
0250  45 44 30 34 30 30 81 60  : 2E
0258  24 45 44 33 46 46 46 29  : DB
0260  82 F0 8F 89 8A FA 89 BB  : 52
0268  82 B7 82 E9 0D 0A 09 2F  : F3
0270  41 09 83 8A 83 55 81 5B  : 0B
0278  83 75 83 47 83 8A 83 41  : 93
-----------------------------------
SUM:  9F 1B 5F 5D 50 A7 EF 65  E2FE

0280  88 C8 8D 7E 28 24 45 44  : 30
0288  30 30 35 42 81 60 24 45  : 21
0290  44 33 46 46 46 29 82 F0  : E4
0298  8F 89 8A FA 89 BB 82 B7  : 19
02A0  82 E9 0D 0A 20 82 C8 82  : 6E
02A8  A8 81 41 82 A2 82 B8 82  : 4A
02B0  EA 82 CC 8F EA 8D 87 82  : 47
02B8  E0 83 75 83 8C 81 5B 83  : 46
02C0  4E 83 4C 81 5B 83 72 83  : 71
02C8  62 83 67 83 60 83 46 83  : 7B
02D0  62 83 4E 82 F0 8D 73 82  : 27
02D8  A2 82 DC 82 B7 0D 0A 00  : 50
02E0  83 75 83 8C 81 5B 83 4E  : B4
02E8  83 4C 81 5B 83 72 83 62  : 85
02F0  83 67 82 F0 82 CB 82 A9  : D4
02F8  82 B9 82 DC 82 B5 82 BD  : 0F
-----------------------------------
SUM:  3E 0F 06 59 1A 67 0E D7  5A23

0300  0D 0A 00 83 8A 83 55 81  : 7D
0308  5B 83 75 83 47 83 8A 83  : AD
0310  41 28 24 45 44 30 30 35  : AB
0318  42 81 60 24 45 44 30 30  : 30
0320  46 46 29 82 F0 8F 89 8A  : C9
0328  FA 89 BB 82 B5 82 DC 82  : 55
0330  B5 82 BD 81 42 0D 0A 00  : CE
0338  83 76 83 8D 83 4F 83 89  : E7
0340  83 80 83 47 83 8A 83 41  : 9E
0348  28 24 45 44 30 31 30 30  : 96
0350  81 60 24 45 44 33 46 46  : 4D
0358  46 29 82 F0 8F 89 8A FA  : 7D
0360  89 BB 82 B5 82 DC 82 B5  : 10
0368  82 BD 81 42 0D 0A 00 53  : 6C
0370  52 41 4D 20 44 49 53 4B  : 2B
0378  20 83 47 83 8A 83 41 28  : E3
-----------------------------------
SUM:  52 66 22 DB A7 10 CA 2A  EBD2

0380  24 45 44 30 34 30 30 81  : F2
0388  60 24 45 44 33 46 46 46  : 12
0390  29 82 F0 8F 89 8A FA 89  : C0
0398  BB 82 B5 82 DC 82 B5 82  : 09
03A0  BD 81 42 0D 0A 00 83 8A  : A4
03A8  83 55 81 5B 83 75 83 47  : 76
03B0  83 8A 83 41 88 C8 8D 7E  : 2C
03B8  28 24 45 44 30 30 35 42  : AC
03C0  81 60 24 45 44 33 46 46  : 4D
03C8  46 29 82 F0 8F 89 8A FA  : 7D
03D0  89 BB 82 B5 82 DC 82 B5  : 10
03D8  82 BD 81 42 0D 0A 00 00  : 19
03E0  00 70 00 26 00 22 00 1C  : D4
03E8  00 22 00 28 00 00 00 00  : 4A
03F0  00 00 00 00 00 00 00 00  : 00
03F8  00 00 00 00 00 00 00 00  : 00
-----------------------------------
SUM:  25 84 62 EC 73 B3 3F 74  8A44
```

リスト10 BREAK BIT CHECKERダンプリスト

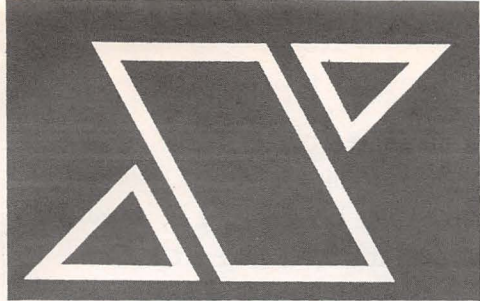
```
0000  48 55 00 00 00 00 00 00  : 9D
0008  00 00 00 00 00 00 02 30  : 32
0010  00 00 00 00 00 00 00 00  : 00
0018  00 00 00 10 00 00 00 00  : 10
0020  00 00 00 00 00 00 00 00  : 00
0028  00 00 00 00 00 00 00 00  : 00
0030  00 00 00 00 00 00 00 00  : 00
0038  00 00 00 00 00 00 00 00  : 00
0040  70 81 93 C9 4E 4F 23 C0  : CD
0048  00 00 00 BA 13 FC 00 31  : FA
0050  00 E8 E0 0D 0C B9 B7 81  : D2
0058  42 0D 00 ED 04 00 67 1C  : C3
0060  45 F9 00 ED 01 00 61 3A  : C7
0068  41 F9 00 00 00 B2 34 90  : B0
0070  70 21 43 F9 00 00 00 BE  : 8B
0078  4E 4F 60 40 45 F9 00 ED  : 68
-----------------------------------
SUM:  3E 2D 16 B3 B7 AF D8 33  E80A

0080  09 00 61 1E 33 FC 07 FE  : BC
0088  00 ED 01 02 41 F9 00 00  : 2A
0090  00 B4 24 D8 34 90 70 21  : 05
0098  43 F9 00 00 00 F7 4E 4F  : D0
00A0  60 1A 41 F9 00 00 01 42  : F7
00A8  43 F9 00 00 02 30 93 C8  : C9
00B0  20 09 53 80 14 D8 51 C8  : 01
00B8  FF FC 4E 75 23 FC 00 ED  : CA
00C0  01 00 00 ED 00 0C 23 FC  : 19
00C8  00 ED 01 00 00 ED 00 10  : EB
00D0  33 FC B0 00 00 ED 00 18  : E4
00D8  13 FC 00 02 00 ED 00 2D  : 2B
00E0  42 39 00 E8 E0 0D 22 79  : EB
00E8  00 00 00 BA 70 81 4E 4F  : 48
00F0  FF 00 4E 75 4E F9 00 ED  : F6
00F8  01 04 00 00 00 00 42 52  : 99
-----------------------------------
SUM:  97 D4 67 EC 7F DA 7F 85  DB47

0100  45 41 4B 20 42 49 54 20  : F0
0108  43 48 45 43 4B 45 52 20  : 15
0110  82 F0 91 67 82 DD 8D 9E  : F4
0118  82 DD 82 DC 82 B5 82 BD  : 33
0120  81 42 0D 0A 09 09 09 42  : 37
0128  79 20 59 2E 4D 69 79 61  : B0
0130  6A 69 6D 61 0D 0A 00 42  : FA
0138  52 45 41 4B 20 42 49 54  : 22
0140  20 43 48 45 43 4B 45 52  : 15
0148  20 82 F0 20 44 4F 43 54  : DC
0150  4F 52 20 82 C6 82 C6 82  : D3
0158  E0 82 C9 91 67 82 DD 8D  : 0F
0160  9E 82 DD 82 DC 82 B5 82  : 14
0168  BD 81 42 0D 0A 09 09 09  : B2
0170  09 09 42 79 20 59 2E 4D  : C1
0178  69 79 61 6A 69 6D 61 0D  : F1
-----------------------------------
SUM:  7E 84 9A 74 37 CD F8 6E  437B

0180  0A 00 60 00 00 9A 00 00  : 04
0188  00 00 83 75 83 8C 81 5B  : E3
0190  83 4E 83 72 83 62 83 67  : 95
0198  82 A8 82 E6 82 D1 81 41  : A7
01A0  83 45 83 43 83 8B 83 58  : 77
01A8  82 CC 83 60 83 46 83 62  : DF
01B0  83 4E 82 F0 8D 73 82 A2  : 67
01B8  82 DC 82 B7 81 42 0D 0A  : 71
01C0  0D 0A 00 83 75 83 8C 81  : 9F
01C8  5B 83 4E 83 72 83 62 83  : 89
01D0  67 82 F0 82 CB 82 A9 82  : D3
01D8  B9 82 DC 82 B5 82 BD 81  : 0E
01E0  42 0D 0A 00 83 45 83 43  : E7
01E8  83 8B 83 58 82 C9 8A B4  : 72
01F0  90 F5 82 B5 82 BD 83 52  : D0
01F8  83 7D 83 93 83 68 82 F0  : 73
-----------------------------------
SUM:  79 CC 9E C1 0D 1C 80 A9  09F6

0200  8E 67 97 70 82 B5 82 BD  : 72
0208  89 C2 94 5C 90 AB 82 AA  : A2
0210  82 A0 82 E8 82 DC 82 B7  : 23
0218  81 42 0D 0A 00 00 48 E7  : 09
0220  FF FE 70 21 43 FA FF 64  : 2E
0228  4E 4F 70 81 93 C9 4E 4F  : 87
0230  41 FA FF 54 20 80 13 FC  : 3D
0238  00 31 00 E8 E0 0D 08 B9  : C7
0240  00 07 00 ED 00 1C 67 08  : 7F
0248  43 FA FF 79 70 21 4E 4F  : E3
0250  70 03 72 00 4E 4F 4A 39  : 05
0258  00 ED 00 FF 67 0E 43 FA  : 9E
0260  FF 84 70 21 4E 4F 42 39  : 2C
0268  00 ED 00 FF 4C DF 7F FF  : 95
0270  00 08 00 22 00 0A 00 1A  : 4E
0278  00 0C 00 0A 00 06 00 3E  : 5A
-----------------------------------
SUM:  5A F9 7A 4D 29 64 39 87  9A74

0280  00 00 00 00 00 00 00 00  : 00
0288  00 00 00 00 00 00 00 00  : 00
0290  00 00 00 00 00 00 00 00  : 00
0298  00 00 00 00 00 00 00 00  : 00
02A0  00 00 00 00 00 00 00 00  : 00
02A8  00 00 00 00 00 00 00 00  : 00
02B0  00 00 00 00 00 00 00 00  : 00
02B8  00 00 00 00 00 00 00 00  : 00
02C0  00 00 00 00 00 00 00 00  : 00
02C8  00 00 00 00 00 00 00 00  : 00
02D0  00 00 00 00 00 00 00 00  : 00
02D8  00 00 00 00 00 00 00 00  : 00
02E0  00 00 00 00 00 00 00 00  : 00
02E8  00 00 00 00 00 00 00 00  : 00
02F0  00 00 00 00 00 00 00 00  : 00
02F8  00 00 00 00 00 00 00 00  : 00
-----------------------------------
SUM:  00 00 00 00 00 00 00 00  0000
```

BASIC

関数を作ってみよう

Izumi Daisuke 泉 大介

だんだんと“プログラミングへの招待”も実践編に入っていきます。今月は，受け取った材料を決められたとおりに処理して製品にする「関数」を扱います。サンプルプログラムやマニュアルをよく見て確認していってください。

さて，X-BASICに触れてみた感想はいかがですか。前回は，X-BASICは数式を計算できること，変数とは数値を入れておく箱のようなものであること，そして，変数に数を入れる方法，数式の中でも変数を使えることを勉強しました。

先月紹介した変数は，数を入れておくための数値変数でした。数値変数には整数を入れておく整数型の変数と，実数を入れておく実数型の変数がありました。整数型の変数を使うには，

```
int a
```

のようにして，整数を入れることができるaという名前の箱を用意します。実数型の変数を使うには，

```
float f
```

のように宣言します。覚えていらっしゃいますか。

さて今月は，X-BASICで扱えるもうひとつのデータ，「文字列」についてやることにしましょう。

文字列を扱う

文字列は画面に「キーを押してください」などとメッセージを表示するときに使います。X-BASICでは「"」でメッセージをくくれば，それは文字列として扱われます。「Hello!」という文字列を作りたければ，

```
"Hello!"
```

とすればいいわけですね。

文字列は，数値と同じようにPrint命令で画面に表示することができます。

```
print "Hello!"
```

とやってみてください。両端につけた「"」が省かれて，画面に

```
Hello!
```

と表示されましたね。

では次に，ちょっとした実験をしてみましょう。

```
print 1
print "1"
```

という2つの命令を実行してみてください。どうなりましたか。「print 1」で表示された「1」の前にはスペースがありますが，「print "1"」で表示された「1」の前にはスペースがありませんね。「print 1」で表示されたのは数値の1です。したがってその左には，符号を表示するためのスペースが空いているのですが，「print "1"」で表示されるのは文字列の「1」ですから，符号がないわけです。

数値を入れる数値変数があるように，文字列を入れる文字列変数がX-BASICには用意されています。数値変数と同じように，文字列変数もまず文字列を入れる箱を用意し，それからこの箱を使うという方法になります。まず，

```
str s
```

で文字列を入れる箱sを用意します。「str」はstring（文字列の意）を略したものです。

```
s="Hello!"
```

で文字列変数sに「Hello!」という文字列を代入し，

```
print s
```

とすれば，画面に「Hello!」と表示されます。

文字列変数には，代入のほかに「+」演算が使えます。これは文字列をつなげる働きをします。

```
print "Good Morning, "+"Friends"
```

とすれば画面には，

```
Good Morning, Friends
```

と表示されますし，先に文字列を代入した変数sを使って，

```
print s+ "Friends"
```

とすれば(半角で)，

```
Hello! Friends
```

と表示されます。

もちろんつなげた結果を再び変数に代入することも可能で，

```
s=s+ "Friends"
```

とすればsには“Hello! Friends”という文字列が代入されることになります。

> チェック !
> 1) 文字列は，"XXXXX"で表す
> 2) str 変数名 は，文字列変数を作る
> 3) 文字列はprint命令で表示できる
> 4) 「文字列+文字列」は文字列を連結する

文字列はその名のとおり文字の集まりです。X-BASICでは全角文字も半角文字も，「"」でくくれば文字列となります。

文字列変数に入る文字数は，デフォルトで半角32文字（全角文字なら16文字）となっています。もっと多くの文字を入れたい場合は，

```
str a[〔文字数〕]
```

のように，変数宣言する必要があります。

便利なブラックボックス，関数

X-BASICで使うひととおりの変数が出そろったところで，X-BASICのプログラミングへじわじわと入っていくことにしましょう。

「関数」というのは，たとえていえば工作機械のようなものです。この機械に原料を与えれば，機械はあらかじめ決められたとおりに動いて，製品を作り出します。X-BASICはこのような工作機械，つまり関数を幾つも持っています。

たとえば「abs」という名前の関数は，数値を原料として受け取り，その絶対値を製品として作り出します。絶対値はabsolute valueといいますが，absはこの略だと考えれば覚えやすいでしょう。

関数は，

　関数の名前（原料の受け口）

という形をしています。「abs」を使うのなら，

```
abs(−10)
```

のように，機械の名前として「abs」を指定し，原料の受け口に「−10」を与えてやればいいのです。absが作り出した製品を見たければ，

```
print abs(−10)
```

とします。なにを与えればなにが得られるのか，それさえ押さえておけば，実際に中でなにが起きているのかは知る必要がありません。関数はブラックボックスとして扱うことができるわけです。

関数を定義するとき，関数を使用するとき，いずれの場合も，関数名に続けて「(」を入力してください。関数名と「(」が離れていると実行できません。

機械の名前のことを「関数名」，機械に与える原料のことを「引数」（「ひきすう」と読む），その製品のことを「関数が返す値」，「戻り値」，あるいは「関数の値」と一般にいいますので，以後これにならうことにします。

関数の値は式の中で使うことができますので，関数が返す値を10倍して表示するなら，

```
print abs(−10)＊10
```

とすればいいことになります。また，関数の値はさらに関数の引数として使うこともできます。

チェック　2
1)　関数とは工作機械のようなものである
2)　関数は，「関数名（引数）」として使う
3)　関数が返す値は式の中で使え，また関数の引数としても使うことができる

自分専用の関数を作る

X-BASICでは，すでに用意されている関数を使うだけではなく，自分で新しい関数を作り出して，それを使うことができるようになっています。「自分が欲しい機能を持った関数を，既存の関数を使って作り出す」ことができるわけです。出来上がった自作関数は，あらかじめX-BASICに用意されている関数と区別なく使用できます。

例として，点A(x1, y1)と点B(x2, y2)間の距離を値として返す関数distanceを作ってみましょう。

ＡＢ間の距離Ｌは三平方の定理より，

$$L=\sqrt{(x2-x1)^2+(y2-y1)^2}$$

で求めることができます。

これをプログラムにするとリスト１のようになります。

40〜80行が関数distanceを作っているところです。自分で関数を作るときは，このように，

```
func 関数名（引数）
  ⌇
endfunc
```

と書きます。関数distanceには，点Aのx,y座標と点Bのx,y座標が必要ですから，引数は４つあります。複数の引数があるときは，「，」で区切って引数を並べることになっています。10行で関数distanceを呼び出していますが，このとき与えた実際の引数が順にx1, y1, ……に与えられます。

あとはこれらの値をもとに，上の式に従って距離を求めればいいだけですね。50行で答えを入れる変数ｄを用意し，60行で距離を計算します。sqrというのは平方根を求める関数で，「√￣」を計算するのに使います。また，ここでは累乗を使う代わりに同じ数を２回掛けています。70行の「return」は，関数が返す値を指定しているところです。計算した距離ｄを返します。そして80行の「endfunc」で，関数distanceは終了です。

40行以降の関数定義ですが，これは略式の書き方です。正確には，

func int distance (x1;int, y1;int,
x2;int, y2;int)

と書きます。関数名の前に，この関数が返す値の型を，引数の後ろには「；」に続けてその引数の型を書くのが正確な書き方です。型がint（整数）の場合は省略できるようになっているのです。引数を小数で受け取りたい場合や，答えを小数で返したい場合には，正確な書き方に従って関数を定義しなければなりません。

関数distanceが整数の引数を受け取り，小数の答えを返すようにしたいなら，

func float distance (x1, y1, x2, y2)

と40行を書き換えることになります。では，このように書き換えて，(0,0)と(100,100)の距離を求めてみてください。

チェック　3

1) X-BASICでは，自分で新しい関数を作ることができる

2) func 型 関数名(引数)～endfuncで関数を定義する

自作関数を支えるアイテムたち

自分で関数を作るには，必要な処理を繰り返したり，条件によって処理を分けたりする必要があります。BASICにはプログラムの流れを自由に変えるための専用命令が用意されています。ここではX-BASICに備わっている，これらのアイテムを簡単なプログラムを作りながら紹介していきましょう。

1．決まった回数だけ繰り返す

たとえば，1から50までの数の和を求める。この問題は私がソロバンを習い始めた小学生の頃，父から与えられました。最初は計算するたびに出る答えが違い，しかも一度として同じ答えが出ないという悲惨な状況だったことを思い出します。

X-BASICには決まった回数だけ繰り返しを行う「for～next」という命令があります。これは，

チェック　4

for 変数＝初期値 to 終値
　命令
next

という書式で使い，「変数を初期値から終値までひとつずつ大きくしながら，nextまでの命令を実行しなさい」という命令です。リスト2を見てください。これは，「変数iを0から10までひとつずつ大きくしながら，print iを実行する」という処理を行うプログラムです。

リスト1

```
10 print distance( 0, 0, 100, 100 )
20 end
30 /*
40 func distance( x1, y1, x2, y2 )
50   int d
60   d = sqr( (x2-x1)*(x2-x1) + (y2-y1)*(y2-y1) )
70   return( d )
80 endfunc
```

リスト2

```
10 int i
20 for i=0 to 10
30   print i
40 next
50 end
```

リスト3

```
10 int i, sum
20 sum=0
30 for i=1 to 50
40   sum = sum+i
50 next
60 print sum
70 end
```

1から50までの合計を求めたいなら，まず答えを入れる変数を用意しておき，for～nextを使って数を足していけばいいですね。プログラムはリスト3のようになります。

このように同じ命令をぐるぐる回って繰り返すことを，ループといいます。

40行は先月も説明したように，「sum＋iをsumに入れる」という意味で,こうしてsumはiの値ずつ増えていくわけです。最初iは1ですから1がsumに加えられ，次にiが2になるので2がsumに加えられ，以下，1から50までが順に加えられていきます。

この簡単なプログラムには，コンピュータでプログラムを書く際のエッセンスが詰まっています。1から50までの数の合計を求めなさい，といわれたときに，for～nextを使えばいいと考えつくにはある程度の慣れが必要です。この連載を通してこのような考え方，センスを磨いていってください。

2．条件が成り立っている間繰り返す

X-BASICで繰り返しを行う際によく使うもうひとつの命令があります。それは,「while～endwhile」です。これは，

チェック　5

while 条件
　命令
endwhile

という書式で使います。条件を表すときは，「>，<，>＝，<＝，＝，<>」が使われ，それぞれ数学で使う「>，<，≧，≦，＝，≠」を意味しています。「代入」と「等しい」が同じ「＝」なので混乱するかもしれません。条件を書くところで「＝」が使われたら，それは「等しい」を意味します。最近のプログラミング言語では，「代入」と「等しい」に別の記号を使

うのが主流になっていますが，X-BASICはBASICの伝統にのっとって（？），同じ記号を使用しています。

whileはリスト4のプログラムのように使います。これは「iが10より小さい間，iをひとつずつ増やしなさい」という意味になります。

また，キーボードから1文字受け取るinkey$を使ってリスト5のようにすれば，スペースキーが押されるまで待つプログラムになります。20行，30行は「inkey$でキーボードから受け取った1文字がスペースでない間，(なにもしないことを)繰り返す」と読んでください。

inkey$はX-BASICが用意している変数のひとつで，押されたキーが入っています。キーがまだ押されていない場合は，キーが押されるのを待って押されたキーを返します。

3. 条件によって処理を分ける

「変数Aが奇数なら“奇数です”と，偶数なら“偶数です”と表示したい」というように，関数を作っていく途中で条件によって処理を分けたいことは少なくありません。これをサポートするのがif命令です。

ifは次のような書式で使います。

```
チェック 6
  if 条件 then 条件が真の処理 else 偽の処理
```

先述した例なら，

if a mod 2 =1 then print“奇数”else“偶数”

となるでしょう。modというのはある数で割った余りを値として返す演算子です。この場合，aを2で割った余りが1なら「奇数」，そうでなければ「偶数」と表示するわけです。

この書式で使うifは，必ず1行で書かなければなりません。 え～！ もっとたくさん書きたいときはどうすればいいの？ という声が聞こえてきそうですね。安心してください。X-BASICは，ちゃんともっとたくさんの処理が書けるifの書式を持っています。

このように，ifでたくさんの処理を書けるのは，X-BASICの大きな特長のひとつです。

このifの書式は次のとおりです。

```
チェック 7
  if 条件 then {
      たくさんの，条件が真のときの処理
  } else {
      たくさんの，条件が偽のときの処理
  }
```

ある数の平方根を表示する関数，pr_sqrtを作ってみましょう。pr_sqrtの引数が負の数だった場合には，最後に虚数を表す「i」をつけて表示することにします。リスト6を見てください。

40～110行がpr_sqrt関数の定義です。pr_sqrtは引数をxで受け取ります。もし受け取った引数xが0以上の数なら，そのままsqr関数を使って平方根を求め，答えを表示します。xが負の数だったら，80行で正の数に変換して平方根を求め，90行で「i」を表示します。80行の最後に「;」がついていますが，これはprint命令に「表示したあと改行するな」と指示しています。このおかげで，平方根を表示したすぐあとに，「i」を表示することができるのです。80，90行はまとめて，

print sqr (－x);“i”

と書いてもかまいません。

1行に書くifは見づらくなりがちです。できるだけ，たくさん書けるifを使うようにしましょう。なお，たくさん書けるifでは，「(もし～なら……，) そうではなくて，もし～なら」という何重にも重ねたIFの処理を書くこともできます。

これは，

```
チェック 8
  if 条件1 then {
    条件1が真の処理
  } else if 条件2 then {
    条件2が真の処理
        ⋮
  } else {
    その他の場合の処理
  }
```

という書式で使用します。

4. 複数の条件を判定する

条件を書くときに，「もし○○で，しかも△△なら」という条件や，「もし○○か，あるいは△△なら」という条件を書きたいことがあります。X-BASICは条件を「and」や「or」でつなぐことによってこれをサポートしています。「かつ」は「and」，

「あるいは」は「or」です。

これを使って「0≦x≦10」という条件を書きたいなら，

0＜＝x and x＜＝10

とします。

今月のおまけ

今月は，簡単な「数当てっこ」ゲームを作ってみました。ルールは，0～5の数を考えて相手より大きい数を考えたほうが勝ちとします。ただし，相手の数より3以上大きな数を考えてしまうと負けです。勘だけが支配するこのゲーム，名づけて「フレンチ勘勘」です（おいおい，年がバレるって）。

今月やったことを駆使して作ってあります。じっくりと読破してみてください。簡単に説明しておきましょう。ゲームは人間対コンピュータで行います。

90～120行は皆さんに数を入れてもらうところで，while～endwhileを使って，0～5の数に限定しています。そのあと130行でコンピュータ側の数を乱数を使って作り，140行で自作のcheck関数を使って勝ち負けを判定します。check関数の返す値が正の数ならあなたの勝ち，0なら引き分け，負の数ならコンピュータの勝ちです。

check関数は300～390行でifの中でifを使うという，頭がこんがらかってしまうような定義になっています。引数aには人間の考えた数が，bにはコンピュータの考えた数が入ってきますので，先の勝敗判定方法と照らしあわせて追いかけてみてください。図を書くとわかりやすいかもしれませんね。

解説していない命令がいくつか入っていますが，これらはX-BASICのマニュアルに載っています。調べてみてください。

それではまた来月お会いしましょう。

リスト4

```
10 int i
20 i=0
30 while i<10
40   i = i+1
50 endwhile
60 end
```

リスト5

```
10 print "スペースキーを押してください"
20 while inkey$<>" "
30 endwhile
40 print "ありがとう"
50 end
```

リスト6

```
 10 pr_sqrt( -4 )
 20 end
 30 /*
 40 func pr_sqrt( x;float )
 50   if x >= 0 then {
 60     print sqr(x)
 70   } else {
 80     print sqr(-x);
 90     print "i"
100   }
110 endfunc
```

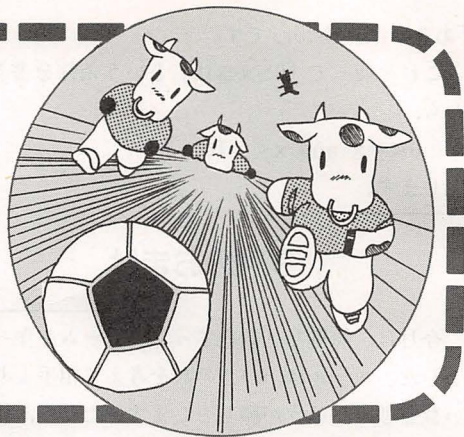
リスト7 フレンチ勘勘

```
 10 /* フレンチ勘勘
 20 /*
 30 int pnum, cnum
 40 int flag
 50 str gamecont
 60 gamecont = "y"
 70 while gamecont = "y"
 80   flag = 0
 90   while flag = 0
100     input "数はなににします? ",pnum
110     if pnum < 6 and pnum >= 0 then flag = 1
120   endwhile
130   cnum = rand() mod 5
140   flag = check( pnum, cnum )
150   color 1
160   print pnum;"-";cnum
170   color 3
180   if flag > 0 then {
190     print "あなたの勝ちです"
200   } else if flag = 0 then {
210     print "ひきわけですね"
220   } else {
230     print "わたしの勝ちです"
240   }
250   input "ゲームを続けますか (y/n)",gamecont
260   if gamecont<>"n" and gamecont<>"N" then gamecont = "y"
270 endwhile
280 end
290 /*
300 func check( a, b )
310   int flag
320   if a>=b then {
330     if a-b < 3 then flag = 1 else flag = -1
340     if a=b then flag = 0
350   } else {
360     if b-a < 3 then flag = -1 else flag = 1
370   }
380   return( flag )
390 endfunc
```

★(で)のショートプロぱーてぃ　その1

がんばれ! 翼くん

Komura Satoshi　古村　聡

お待ちかね，Oh!X一番の人格者といわれる(で)氏の登場です。彼は，編集部に届きながら日の目を見る機会の少ないショートプログラムを紹介する場を作ろうと，ついに立ち上がりました。読者の皆さん，ぜひ力強い応援をお願いします!

あーっ!　めでたい，めでたい，ほんとにめでたい!　どーもっ,2カ月間出番がなくて，ヒマでヒマで死んでしまいそうだった(で)です。先月のゲームレビューの片隅に予告したとおり，今月からめでたくショートプログラム関係の(関係っていうのがちょっと怖い)連載が始まります。さ，ぱぁーっといくぞ，ぱぁーっとっ!

さて，この「ショートプロぱーてぃ」でなにをするかというと，まぁ，このコーナーでは基本的には投稿されてきたショートプログラムを発表する場なわけです。

編集部には毎月いろいろなプログラムが投稿されてきますが，ショートプログラムを発表する機会がなかなか作れない。で，私がそんなショートプログラムの救済をしよう，と立ち上がったわけです。え，カッコいいじゃないかって，そりゃどうも。

で，あらためて投稿歓迎するわけなんですけど，はっきりいってなんでもありです。もうGod祝社長[1]じゃないけど何の挑戦でも受けちゃいます。MZシリーズにXシリーズ，BASICやマシン語(この場合はアセンブラのソースリストをつけてね)，C，Pascalなどなど。もちろん，これ以外でも面白ければ載っけちゃいます(なんせturboLOGOなんて言語もありましたからねー。Xシリーズで動くんだったら載せないわけにいかないですから)。

ただ，選考基準は私が面白いと思うかどうかなので，そこんとこご了承ください。たとえばHuman68kやS-OSなどのDOSのユーティリティプログラムやツールなんかでも，私が「OK!」と思ったら載せるだろうし，原稿を読んでいて「これ，楽しいなぁ」と感じただけでも載せることがあるでしょう。要するになんでもありなんです。イラストでもお便りでもプログラムでも。あ，最近見つけた毒物飲料(ネ○○ルのコーヒースカッシュとか，なぜか小田急の売店でしか売ってない麦コーラとか)なんかも大歓迎です。そしたら荻窪さんとか編集さんあたりに飲ませて楽しみますんで(あ，胃を押さえて逃げないでくださいよ!)。

で，もし気に入るプログラムがなかった場合はどうなるか。そんなときは私自身がプログラムを組むことになるかもしれないし，掲載まであと1歩のプログラムに手助けをして載っけたりもするかもしれない。さらにはプログラムについてきた手紙だけ紹介したり，私が言いたい放題したり，場合によっては私の考えているもっとも恐ろしいコーナー(さぁなんでしょう?　フフフのフ，と)になってしまうかもしれません。

ですから，そうならないためにも，じゃんじゃんショートプログラムを作って送ってください。

はーっ，とてつもなく長い前ふりを終えたところで今月の作品いきます。京都府の川島裕一さんからMZ-2500用/BASIC-M25のプログラムです。

「本当はもっと早く出すつもりだったのですが，そこは受験生の悲しさ，差し止めをくらい愛機に触ることさえ許されなかったのです。しかしっ!　このたび,無事合格しましたので，早速投稿させていただきました」……あ,復帰第1作，おめでとうございます。うーん，連載開始にちょうどいいめでたさだ。どうせだからA/B両面とも(そう，彼はカセットテープでプログラムを送ってくれたのだ)掲載してしまおう。ではどうぞっ!

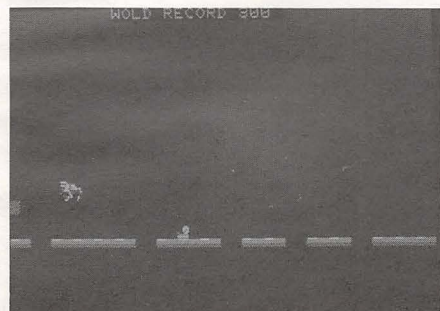
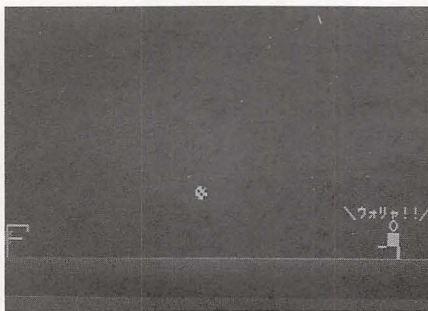
★Wormy Highway

川島　裕一　(京都府)

ストーリー：愛車で高速を飛ばしていると，突然，目の前に虫が食ったような穴だらけの道路が現れた。なんと，ピンク色の謎の物体カジカジが，道路を食い散らかしているのだ!「このオレにケンカを売っているのか!?　許さんっ!」……かくして君は果てなきバトルの入り口へと向かって行くのだった……(なんのこっちゃ)。

遊び方：RUNすると数字を聞いてきますので，－32768から32767までの数字を入力します。するとゲームが始まりますので，次に挙げたキーで自分の車を操作してください。車がカジカジにぶつかったり穴に落ちたりすると，そこでゲームオーバーとなります。

[スペース]……押して前進，離して後退
[8]……………大ジャンプ
[5]……………中ジャンプ
[2]……………小ジャンプ

なお，変数が何を指すかはプログラムリストを見てください。

Wormy Highway

Soccer

★Soccer

同じく**川島　裕一（京都府）**

ストーリー：なし。（こら，こらっ！）

遊び方：RUNするとすぐにゲームが始まります。サッカーボールを蹴って，うまくゴールに入れると得点できます。ただ蹴飛ばすだけだとボールは大きくそれてしまいますので，スペースキーをたたいてボールの軌道を調節してください。これを10回やると得点に応じてコメントが出てきます。回が増すごとにボールの初角度が上がっていきますのでうまく調節しましょう。なお，REPEAT ON4としてからゲームをするとドライブシュートが楽しめます。

……うーん，このあいだのBASICの特集で「がんばれ！　カズシゲ君」っていうのを荻窪さんが作ってたけど，このサッカーゲームは，なんとなく「がんばれ！　翼くん」っていうノリですねー。そういうタイトルつけると，漫研の女の子たちに受けそうだなー。

ともあれ，どちらもなかなか面白いゲームでした。Wormy Highwayではハイウェイを3階建てにするとパックランドみたいでいいなーとかいろいろ思ったんだけど，ゲーム自体は私はいじってません（リストは少々いじりましたが)。変数表もあることですんで，そのへんは皆さんでやってみてください。

それでは今月はこのへんで。謎の投稿第2号，第3号を求めますので，どんどん参加してください。たくさんの快作を待ってますよ！

1)　だから，"Oh, My God!"を私が訳すと「きゃあ祝さん助けて！」……になるわけないでしょ，もう。

リスト1　Wormy Highway

```
  100 '(**********************************************************************)
  110 '(* H=world record        L=ｸﾙﾏﾉｽｽﾝﾀﾞｷｮﾘ                                 *)
  120 '(* T=ｶｼﾞｶｼﾞﾉｶｳﾝﾄ                                                      *)
  130 '(* x,y=ｸﾙﾏﾉｻﾞﾋｮｳ                                                      *)
  140 '(* x1,w=x,yﾉｿﾞｳﾌﾞﾝ                                                    *)
  150 '(**********************************************************************)
  160 INIT "CRT:40,25,0":KMODE:KLIST 0:GOSUB 240:M$="@M2@19T255L3201A":H=300
  165 CU1$=CHR$($1F)+CHR$($1D)+CHR$($1D)
  170 INIT "CRT2:":CLS:INPUT "LVEL No? ",B:RANDOMIZE B:CGEN 1:CLS:LOCATE 0,18:PR
INT STRING$(19,"fg"):REPEAT ON ,4:T=38:X=5:Y=15:W=0:L=0:CGEN:LOCATE 10,0:PRINT [
6] "WOLD RECORD";H:CGEN 1:C$(0)="`b"+CU1$+"ac":C$(1)="`b"+CU1$+"de":R$(0)="fg":R
$(1)="fg"
  180 LOCATE 0,18:PRINT CHR$(8):IF A=1 THEN LOCATE 38,18:PRINT R$(RND*A*2)
  190 LOCATE X,Y:PRINT "  "+CU1$+"  ":X=X+X1:W=W+1:Y=Y+W:LOCATE X,Y:PRINT C$(A)
  200 K$=INKEY$:K=VAL(K$):S1$=SCRN$(X,Y+2,2):S2$=SCRN$(X+2,Y+1):POKE@ $38,T+680,
32:T=T-.5:POKE@ $38,T+680,105:IF T<1 THEN LOCATE T,17:PRINT " ":T=38
  210 IF Y>18 OR S2$="i" THEN PLAY INIT:PLAY ,,"@M2@10AB":CGEN:LOCATE 10,10:PRIN
T [5] "YOUR RECORD..";L;"m":PAUSE 20:IF L>H THEN H=L:GOTO 170 ELSE 170
  220 DEF CHR$(105)=B$(A):IF S1$<>"  " THEN W=-1+(K=2)*3+(K=5)*4+(K=8)*5
  230 X1=(K$="" AND X>5)-(K$=" " AND X<20):A=A XOR 1:PLAY ,,M$:L=L+1:GOTO 180
  240 F$=STRING$(16,"F"):FOR I=1 TO 9 :READ D$:DEF CHR$(95+I)=HEXCHR$(D$):NEXT:F
OR I=0 TO 1:READ D$:B$(I)=HEXCHR$(F$+D$):NEXT:DEF CHR$(32)=HEXCHR$(F$+STRING$(32
,"0")):RETURN
  250 DATA FFFFFFFFFE8ED9D00000000F01FFE7E00000000F008ED1D,99BC021A050503877E7F
FF9F377F3000193C021A05050000,FFFFFFFFF3FDFEF0000000000C0E070000000000000C0E0,FF
030108050783C73EFCFFCF96BC1000F20201080404000,99BC02020105079F7E7FFF87637F1C181
93C020201050418
  260 DATA FF030100010787CF3EFCFFC7AA942C08F202010000040408,FFFFFFFFFFFFFFAA7E00
000000000000FFFFFFAA55000000,FFFFFFFFFFFFFFAA0000000000000000FFFFFFAA55000000,FF
FFFFFFFFFFFFAA7E0000000000000000FFFFFFAA55000000,060F8B7E5603A9FF060F0B065000A800
  270 DATA 060F0D0B867FABFF060F0D0B0600FE00
```

リスト2　Soccer

```
   10 '(********************************)
   20 '(*c=ballﾉｶｸﾄﾞ                   *)
   30 '(*i=kickﾉｶｲｽｳ                   *)
   40 '(*s=shootﾉｶｲｽｳ                  *)
   50 '(*w=ballﾉyｻﾞﾋｮｳﾉｿﾞｳﾌﾞﾝ          *)
   60 '(*x,y=ballﾉｻﾞﾋｮｳ                *)
   70 '(*c$=MYｷｬﾗ                      *)
   80 '(*m$()=ﾒｯｾｰｼﾞ                   *)
   90 '(********************************)
  100 CU1$=CHR$($1F)+CHR$($1D)+CHR$($1D)+CHR$($1D)
  110 CU2$=CHR$($1F)+CHR$($1D)+CHR$($1D)
  120 INIT "CRT:40,25,0":KLIST 0:COLOR=(1,0):COLOR ,,9:KMODE:DIM X%(18),Y%(18):C
$(0)="0 "+CU2$+"█>"+CU1$+" |￣":C$(1)="0 "+CU2$+"█>"+CU2$+"|\":C$(2)="0 "+CU2$+"█
 "+CU2$+"||":C$(3)="0 "+CU2$+"█ "+CU1$+"/|":C$(4)="0 "+CU2$+"█ "+CU1$+"￣|"
  130 FOR I=0 TO 18:READ A$:X%(I)=VAL("$"+A$):NEXT
  140 Y%(0)=8:Y%(1)=8:FOR I=0 TO 5:READ M$(I):NEXT:FOR I=0 TO 2:LOCATE 0,20+I:PR
INT [2] STRING$(40,"█"):NEXT:LOCATE 0,17:PRINT "┌─┐"+CU1$+"├─"+CU2$+"|":C=.08:I=
0:S=0
  150 I=I+1:C=C-.005:X=280:Y=152:W=-3:CLS 2:PUT@ (X,Y),X%,XOR:FOR L=0 TO 4:LOCAT
E 36,17:PRINT C$(L);:PAUSE 1:NEXT:LOCATE 32,16:PRINT "\ｳｫﾘｬ!!/":PLAY ,,"@M2@11L1
006A"
  160 W=W+C:ON ERROR GOTO 180:IF INKEY$=" " THEN W=W+.5
  170 PUT@ (X,Y),Y%:Y=Y+W/.8:X=X-7:IF X>2 THEN PUT@ (X,Y),X%:GOTO 160
  180 IF Y<165 AND Y>137 THEN LOCATE 10,1:PRINT "NICE SHOOT!!":PLAY ,,"@M2@1004A
":S=S+1 ELSE LOCATE 10,1:PRINT " NO  GOAL !!":PLAY ,,"@M2@2304A"
  190 PAUSE 10:LOCATE 10,1:PRINT SPC(12):LOCATE 32,16:PRINT SPC(8)
  200 IF I<9 THEN 150 ELSE LOCATE 10,10:PRINT [2] M$(S/2):PAUSE 10
  210 DATA 8,8,3C00,C0C,7E0C,4E4E,FF4E,3939,FF39,3939,FF39,E7E7,FFE7,E7E7,7EE7,1
C1C,3C1C,1C1C,1C,"ｷﾐﾆﾊ ｻｯｶｰ ﾊ ﾑｲﾃﾅｲﾝﾞｬﾅｲｶﾅ･･･","ｵｲ ｵｲ ｷﾐﾊ ｱﾐﾊﾞｶｲ?","ｱｼ ｦ ﾈﾝｻﾞｼﾀﾉ
ｶﾅ?","ｶﾝ ｶﾞ ﾆﾌﾞｯﾀﾉｶｲ?","ｴｰｽｽﾄﾗｲｶｰﾏﾃﾞ ｱﾄｽｺｼ","ｷﾐﾊ ﾓｳ ﾂﾊﾞｻｸﾝﾀﾞ!"
```

CGA初心者救助隊出動！

BGMはサンダーバードがいいな

プロジェクトチームDōGA　かまた ゆたか

DōGA・CGAシステムは，CGAをやりたくてもできなかった人にとって良いきっかけになるでしょう。しかし“マニュアルだけではさっぱりわからない”と投げ出してしまう人もいるかもしれません。今回は初心者救済のためのCGAシステム入門を行います。

前回発表したDōGA・CGAシステムは，もうお手元に届きましたか？　えっ届いていない？　そりゃそうだ，まだ発送していないのですから。

だって発送は8月からっていったじゃありませんか。ただ今最後のチェックに追われているところですが，どうぞご安心ください。発送の準備はちゃくちゃくと遅れております（コマッタモンダ）。

そんなわけで，今回はマニュアルとはひと味違った入門講座にアプローチしてみました。

初心者よ，あきらめるでない！

このCGAシステムを手に入れるほとんどの方は初心者だと思います。3Dグラフィックなんて，プログラミングはおろか，その手のソフトにも触ったこともなければ，専門書だって読んだことのない人も多いでしょう。

もちろん，CGAシステムは3D図形を扱うために必要な数学力やプログラミングの知識がなくても使えるようにはなっています（Oh!Xの3Dグラフィック特集は難しくってわかんなぁいという人でも大丈夫)。とはいっても，このシステムを手に入れた人がすぐさまCGアニメーションを自由自在に作れるというわけにはいきません。

たとえばワープロの場合，多くの人はそれがどんなもので，どんな機能があるのか，およそ予想がつくし，似たようなソフトを使った経験があるため，いまさら“難しい”という印象は持ちません。が，初めてワープロというものを知ったときはどうだったでしょうか。なんの知識もなく，見当もつかないあなたにとって，ワープロはものすごく難しいものに見えたでしょう。

CGAも同じです。CGA制作自体は決して難しいものではありません（少なくともワープロよりは簡単だと思います)。ただ，見当がつかないので，なにやら難しいものに見え，そしてひと通りできるようになるまでの間は，難しく感じるのです。

そして，今の段階でCGAに必要なことは，知識でも，センスでもありません。努力と根性です。まず，マニュアルはちゃんと読んでください。初心者なら，“自分はCGAの基礎知識がないのだ”と謙虚になって，少なくともマニュアルぐらいは隅から隅まで目を通しましょう。行き詰まっていたことが，実は単なる思い違いだったなんてよくあることです。

また，これを機会にCGの専門書を1冊読むことをお勧めします。その後にマニュアルを読むと，ずっと理解度が増すかもしれません。それでもだめだったら，当チームの「あき姫」にお手紙ください。あの「あき姫」がまともに答えてくれるとは思えませんが，この連載中に質問の特集をすることも検討しています。

パソコンの基礎知識を身につけよう

それでは，実際の作業に入る前に，フォーマットされたデータディスクを用意して，サンプルディスクの中の適当な形状データをコピーしてください。となるところなのですが，もしかしたら，フォーマットやコピーの仕方がわからない人もいるのじゃないか？　いやっきっといるでしょう。

実は，以前このシステムの試作版を一部の人に配布したことがあるのですが，そのとき，

「マニュアルはディスクの中に入っていますので，エディタで見てください」

「プログラムのソース，ご希望の方には公開します」

と，書いておいたところ，

「マニュアルの見方が，わかりません」

「ソースってなんですか？　でも，ください」

というお便りが何通か来てしまったのです。

X68000を買って，パソコンの勉強はちっともせずにゲームばかりしている人，この場で懺悔しなさい（オット，うちのスタッフも何人か懺悔している～う)。

それでは，勉強を始めましょう。まずは「Human68kユーザーズマニュアル」をじっくり読んでみましょう(実は私も今まで開いたことがなかった)。うげ～，こんなもん読んでられっかよ～(5分と続かなかった)。

え～，コホンッ。この手のマニュアルは全部読む必要はありません。必要なところだけ読めばよいのです。なんと驚いたことに，とりあえず使えるようになるのに必要なところは十数ページしかないのです。

まず,「2.2 Human68kの起動」で, コマンドモードとカレントドライブの変更を理解してください（それ以外はどうでもいいです)。CGAシステムでHuman68kのコマンドモードになるてっとり早い方法は, システムを終了すればよいのです。マウスの右ボタンを押せばメニューが現れますので, ボタンを押したまま下にずらして, 終了のところで離すだけです。

「2.3ディスクのフォーマット」を読めば, フォーマットの仕方がわかります。そしてメインは「第3章　ファイルとディレクトリ」です。この12ページはしっかり読んで, 各コマンドを理解してください。これだけ知っていれば, とりあえず十分です。わかりにくければ（破壊してもよいようなディスクを使って）実際にいろいろやってみてください。

CGAシステムの機能をフルに使いこなすためには, 後後エディタやOSの知識も必要になってきます。暇を見つけて, より幅の広い知識を取得してください。

全自動CGA

さて, CGAシステムには, 手始めになんでもいいからアニメーションを作ってみたいという方のために, AUTOという全自動CGAのコマンドが用意されています。今回はこのAUTOコマンドを使用して, 簡単なアニメーションの作り方を説明しましょう。

まずは, フォーマットされた新しいデータディスクにサンプルディスクの中の形状データをどれか1つコピーしてください。DIRでファイル名を表示してみて, 拡張子が.SUFとなっているファイルが形状ファイルです。データディスクが準備できたら, CGAシステムに入り, AUTOを先ほどの形状データを指定して, 実行してください。操作はそれだけで終わりです。

初心者のために, 実行方法を丁寧に解説すると以下のようになります。

1) 形状データの入ったデータディスクを用意する。
2) CGAシステムディスクをAドライブ, 上記のデータディスクをBドライブに入れてリセットする。しばらくすると, CGAシステムが立ち上がる。
3) 「形状ファイル」と書かれているところをクリックする。形状ファイルのウィンドウが現れ, サンプルディスクからコピーした形状データが表示される。
4) マウスの右ボタンを押し, メニューを出す。押したままコマンドに持っていって, そこで離すとコマンドメニューが表示される。
5) コマンドメニューのAUTOをクリックする。AUTOの実行ウィンドウが現れ, 形状データのウィンドウも反転する。
6) サンプルディスクからコピーした形状データをクリックする。そのデータが反転する。
7) AUTOの実行ウィンドウの「実行」をクリックする。
8) 動画が1枚1枚作画された後, アニメーションになるまでひたすら待つ。
9) アニメーションにあきれば, ESCキーを2回押して終了する。メインメニュー（PES）に戻る。

このAUTO実行時に, 画面にいろいろなメッセージが表示されますから, 一度読んでみてください。たとえば, 形状データとして「CAR.SUF」を指定した場合,

形状ファイル：CAR.SUFを読み込みます。
アトリビュートファイル：CAR.ATRを読み込みます。
アトリビュートファイルがありませんので, ファイル名「CAR.ATR」でアトリビュートファイルを作ります。
フレームファイル：CAR.FRMを読み込みます。
フレームファイルがありませんので,
⋮

(ちなみに, この原稿を書いている時点では, まだAUTOのプログラムは完成していないので, メッセージの内容は多少異なると思われます)

AUTOがしていることは, 形のデザインだけが終了している状態から, 色のデザインを行い, 動きのデザインをし, フレームファイルを作り, 作画し, タイムチャートファイルを作って, アニメーションする。つまり, 人間がしなければいけない作業を, 順番にやってくれているわけです（CGAシステムのマニュアルの「CGA制作の流れ」で勉強してください)。

あき姫の迷える子羊のコーナー

こんにちは, あき姫です。某月某日, 風邪をひきました。実は姫は風邪の長引く体質で, 2週間以上たった今でもまだ鼻と喉が本調子ではありません。今年の風邪は, 姫に限らず長引くようなので, 皆さん気をつけてください。

今回は, 最近DōGAに寄せられたお手紙の中の質問(?)を取り上げてみました。

子羊：(ある手紙の途中)……ところで, 先日たまたまお寺へ行ったとき, 和尚さんが話しておられました。“この世のすべてのものに, 神様はいらっしゃる”と。コンピュータの神さまってどんなんでしょうね？

姫：あっ知ってます。コンピュータの神様は, と～っても小さくて, コンピュータの中を0と1の旗を持って走り回っているんだそうです。姫も実際に見たことはないのですが, 聖域（コンピュータの中）に手を触れて, ピシッと天誅を受けたことがあります。とっても痛かったです。

＊

子羊：ぜひ, 御社のCGAシステムの独占販売について, 当社と契約を結んでいただきたいのですが……。

姫：たっったわけものっっっ！　町内引き回しのうえ, 打ち首, 獄門！

＊

子羊：CGAシステムの作画はとても早いですね。スキャンラインにおける, 半透明ポリゴンのアンチエリアシングの処理がよくわかりません。簡単に解説してください。

姫：姫には, なんのことだかさっぱりわかりません。詳しく解説してください。

おおっ，なんと便利なツールでしょう。しかし便利な反面，限界もあります。自動的にということを裏返せば，色のデザインはデタラメであり，動きのデザインはワンパターンなのです。

CGAアーチストへの第1歩

残念ながら今のあなたは，デザインを全部プログラムにやらせているので，CGAアーチストとは呼べません。しかし，あせらず，少しずつ前進しましょう。

AUTOを使ってアニメーションすると，まずこのデタラメな色使いをなんとかしたくなるはずです。方法は2つあります。

1) サンプルディスクの中から，先ほどコピーした形状ファイルと同じ名前のアトリビュート（拡張子.ATR）をコピーする（AUTOはアトリビュートファイルを自動的に作りますが，ディスクの中にすでにアトリビュートファイルが存在していると，そちらを優先します)。
2) ATR（アトリビュートエディタ）を使用して，自分で色のデザインをする。

アトリビュートエディタの使い方はマニュアルをよく読んでください。最初はデタラメでもかまいません。慣れてくれば，あなたのセンスを生かせるようになるでしょう。着々とCGAアーチストに近づいていますね(よかったよかった)。

次に，ワンパターンな動きをなんとかしましょう。これはAUTOのオプションを利用します。オプションとは，そのコマンドを実行するときの条件で，この場合，AUTO実行時に“たまには他の動き方をやってみんかい”という条件をつけるわけです。

メインメニュー（PES）でAUTOをクリックしたときに現れるAUTOの実行ウィンドウを見てください。右下のほうに「オプション」と表示されているはずです。そこをクリックすると，実行ウィンドウの右側にオプションウィンドウが現れます。

また，AUTOの実行ウィンドウの下方にある「解説」をクリックすると，AUTOのマニュアルが出ます。複数のページになりますので，「次ページ」で続きが見られます。そうやってオプションを探してください。

これまではオプションをなにも指定していなかったので，Z軸回りに回転する動きでしたが，オプションによってY軸や，X軸回りに回転することもできます。少し変わったところで，一直線上（X軸上）を動かすこともできます。

解説ウィンドウと同じオプションがオプションウィンドウにも並んでいますので，好きなものを選んでクリックしてください。続けて，その右側の空欄にキーボードから“20”を打ち込んでください。これは動画の枚数で，少なければ速く，多ければゆっくりと動きます。そして「実行」です。さっきとは違ったアニメーションができるはずです。

このようにして，次回までの1カ月間は優に遊べるでしょう（んなわけないか)。やっぱり，動きがワンパターンだというのなら，FFE（フレームファイルエディタ）で本格的な動きのデザインにも挑戦していってください。

AUTOはCGAシステムのほんの一機能でしかありません。このシステムはもっともっと奥の深いものです。ですから，初心者の方は一度に全部習得しようとせず，AUTO，ATR，FFEと一歩一歩階段を上って行けばよいのです。

メインメニュー（PES）の使い方はひと通り理解できたはずですから，他のプログラムも少々使い方がわからなくてもどんどん使ってみてください。あなたが早くこ

ただいま，会員募集中

C-COM（中部コンピュータクラブ機構）

代表　大杉 健一

新設コラムです（毎月コラムを新設していると，そのうち本文がなくなるのではないだろうか)。タイトルにありますように，全国で活動しているアマチュア団体を紹介します（本当に会員募集中とは限らないけど)。べつにCGA専門のチームでなくてもよいでしょう。今回紹介いたします「C-COM」も，コンピュータ全般の活動をしています。

このコーナーに掲載を希望されるチームは，遠慮なくお手紙ください。

1) C-COMとは

C-COMは中部地区を中心とした大学，高校，各種専門学校等のコンピュータクラブの集まりです。非営利を原則とし，目的は構成団体間の交流，各団体の活動環境の整備，一団体では困難な企画活動の支援などです。

CGAも一団体だけではなかなかよいものが作れないので，団体間で協力していこうと考えています。

2) CGAの計画

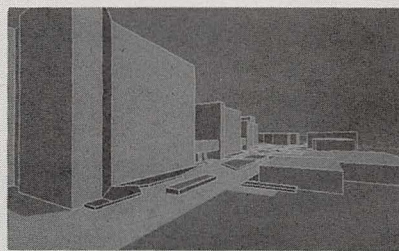
現在では各団体個別で作品を制作し，イベントなどで発表しています。C-COM全体としては，DōGAのCGAシステムをベースとしてデータの共通化を図るとともに，ワークステーション等の使用も考えています。

3) 掲載作品（口絵参照）の解説

これは参加団体のひとつである豊橋技術科学大学コンピュータクラブ（TCC）が制作いたしましたCGAの1カットです。

4) 構成団体

現在の構成団体は18団体です。C-COMではCGAに関わらず参加団体を募集していますので，参加を希望される団体はご連絡ください。

コラムの第1回を飾るにふさわしい，今年設立したばかりの団体です。参加は団体単位ということですが，詳しいお問い合わせは下記の住所にお願いします。

豊橋技科大をモデルにしたCGです。建物はすべて面で表現されていて，これについては，独自に考えた優先順位法で対処しています。
豊橋技術科学大学コンピュータクラブ
CG班　山川大介

〒466　愛知県名古屋市昭和区川名町1-23-1
タウンハウス川名 102号室　C-COM本部

のCGAシステムを使いこなせるようになることを期待します。

来月はモデリングだーっ

さて，今回は初心者対策ということで，入門の入門に徹していましたがいかがだったでしょうか。CGAシステムを手に入れはしたが，手が出せないといった方のお役に立てれば幸いです。

次回はこのCGAシステム最大の難関，難攻不落の宇宙要塞，3Dモデリングツール（CAD）を攻めてみようと思います。使いこなすテクニックなど，少し実戦的な話になってくるでしょう。予習はしっかりやっておいてください。

ところで，このCGAシステムを配布すると発表してから，身辺が妙に騒がしくなってきました。TVや雑誌で紹介されるわ，ネットでも噂が流れるわ。それにつれて，問い合わせなども増えています。はっきりいっておきますが，私達はプロのソフトハウスではありません。CGAの普及を目的に活動しているアマチュア団体です。勘違いせぬようお願いします。

わ〜い，わ〜い，にこ・にこ・ぷん！

★DōGA・CGAシステムは一般のお店では取り扱っていません。私達の活動に賛同してくださるアマチュアの方には，カンパ（1口：1000円より）と実費（3000円）だけで配布しています（プログラムは無料です）。郵便振替で申し込んでください。

申し込み期間：1989年7月1日〜10月30日

郵便振替口座：大阪 3-109598　口座名：鎌田　優

または，DōGAプロジェクトルームCGAシステム配布係に直接現金書留で申し込んでいただいても結構です。

なお，発送は申し込まれた順番に行いますので，場合によっては少し遅れることがあります。

★CGAシステム，本連載，各コラムについてのお手紙お待ちしています。

〒533 大阪市東淀川区淡路5-17-24 篤コーポ102号室

DōGAプロジェクトルーム「なんでもどんとこい」係

●寺田の教育的指導　その2「驚異！　始動不能！」●

私の野望は一瞬にして崩れ去ってしまった！　先月号で「皆さんの作品が送られてきたら……」などと言っていたのは，よくある楽観的見通しであるということは，言うまでもありません。し，しかし，……ようするにやね，まだシステムを配布もしてないのに作品が送られてくるはずがないという当然の事実に気づくのに人一倍時間がかかってしまったわけです。なにしろCGAシステムは日常的に使っているので，こんなものはあって当然というような大それた意識を持ってしまっていたのです。

「本文」を書いている某Kさんも焦ってはるみたいやけど（他人事やないぞ！）まあ来月には皆さんのお手元にめでたくCGAシステムが届くことでしょう。そうしたら今度こそどんどん作品を送って私を楽にしてください。

というわけで今回は自分を自分で指導するという人類始まって以来の偉業に私は挑戦することになり，急いで1カット作りました（しかし「急いで」CGAが作れてしまうとは！　やはりCGAシステムは偉大である）。

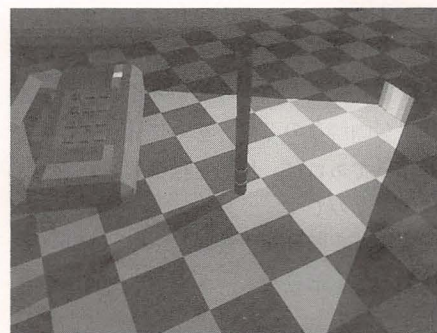
＊

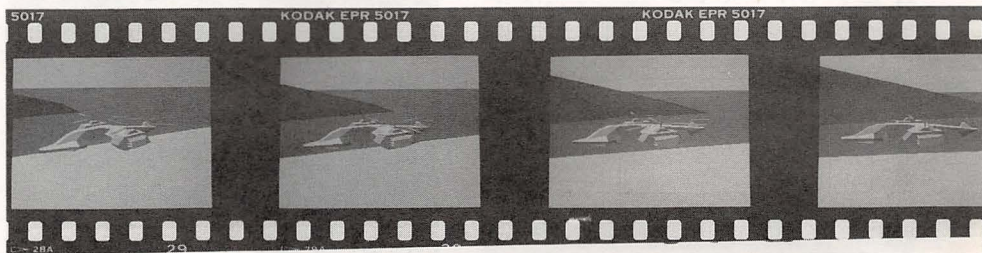
皆さんこんにちは，教育的指導のコーナーです。今日は大阪府の寺田さんの作品を見てみましょう。

おっ，こ，これは，ヘリがローターを回しながら飛んでいるではないか！　そうかわかったぞ！　構造体を隠し味に使っているんだぁ（私は隠れ「ミスター味っ子」ファンではありません）。

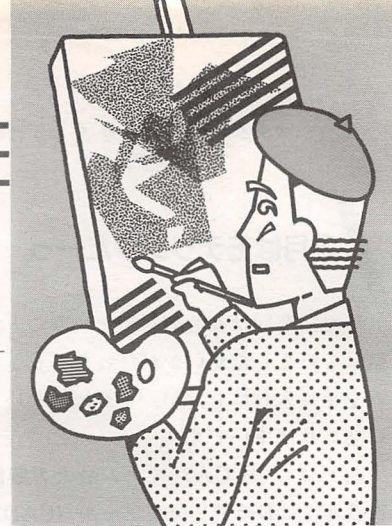
さて構造体とはなんでしょう？　CGAシステムで使う構造体というのは，イメージ的にいうと，『♪親亀の上に子亀を乗せてー，子亀の上に孫亀乗せてー，』という感じです（なんのこっちゃ！）。つまり，子亀は親亀から振り落とされない限り，自分は動いたつもりがなくても，親亀が動けばつられて一緒に動いてしまうということです。

ちょっと無理がある説明ですが，これをこのヘリコプターにあてはめると，ヘリの機体が親亀で，ローターが子亀にあたります。ローターは機体に対しては静止していて，ただその場でクルクル回っているだけですが，けれども機体が動けばちゃんとそれについてきます。もしここで構造体を使わないと，機体は飛んで行くのに，ローターだけ取り残されてひとりでクルクル回るという大ボケなことが起こってしまいます。

おわかりいただけたでしょうか？　このように構造体は物体の「可動部分」を表現するときにたいへん便利なのです。詳しい使い方はマニュアルを見てください。わかりやすく解説してあるはず(?)です。構造体を使いこなせるようになれば，いろいろ面白い表現ができるので皆さんも勉強してください。

さて，右上の絵は，最近仕様書の片隅から発見されたRENDの隠し機能である点光源をフルに使っています。従来の平行光線との違いがおわかりでしょうか？　なかなか綺麗でしょう。これは当然，「技あり！」ですね。この点光源については，また機会を改めてお話したいと思います。では，さいなら。

MZ-2500用グラフィックエディタ作成講座〈2〉

ラインの応用とGCRTCの使い方

Motohashi Jun
本橋　純

任意領域のコピー機能やアナログRGBを最大限に生かすグラデーション，必須ともいえるルーペ機能。今月の拡張で画餅も一人前のグラフィックエディタになります。追加部分を打ち込むだけの簡単拡張でお届けしましょう。

今月はパレットウィンドウ，ルーペウィンドウ，そしてコピー機能が登場します。どれも存在意義のあるものばかりですので，ぜひ入力してください。

今後も画餅には多くの機能を拡張していきますが，画餅は特に多機能さだけを目指したグラフィックエディタというわけではありません。もともと，ウィンドウまわりや基本となる描画ルーチン群を核に，どうしても必要なものから順に強化していった結果が画餅なのです。基本部分さえしっかり作ってあれば，機能を拡張していくことは簡単なのです。

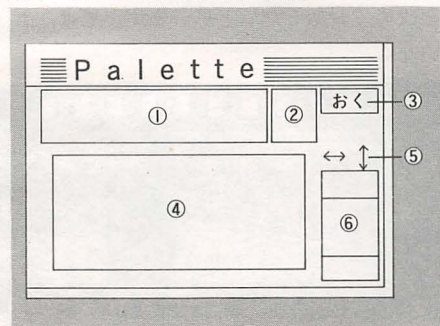
では，今回はやや貧弱だったエディット機能の基本部分をもっと充実させておきましょう。

追加部分入力方法

とりあえず，先月の基本部分に今回解説するプログラムを加えていきましょう。手順は以下のとおりです。

1)　マシン語エリアを確保し，

　BLOAD“画餅AMA-25h”, adr

で読み込みます。adr は入力したときの開始番地です。MACINTO-Cなどを使ってアドレスをずらして入力したときに必要なものです。

2)　今月の掲載プログラムを入力します。

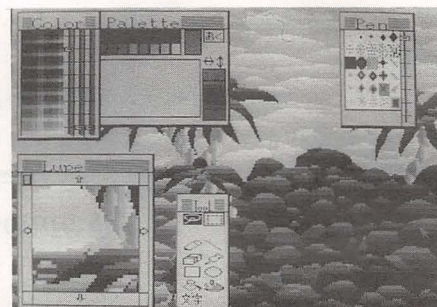
3)　変更前のプログラムをバックアップしておいたうえで，前回と同様にセーブします。

これで，描画に最低限必要なものはすべて揃ったはずです。まだファイル関係がサポートされていないのでグラフィックエディタとしては未完成ですが，実用上必要十分な機能とはこんなところではないでしょうか。

操作方法

1)　Paletteウィンドウ（図1）

①の色見本の部分を左クリックで選択します。現在のペンの色は②の部分に表示されます。しかし，これはリアルタイムに表示しているのではないため，スポイトやカラーウィンドウで色を変更した場合でも変化してくれません。

③で左クリックしたあと，①の色見本や⑥のグラデーションの上下の色の位置で左クリックするとペンの色が置かれます。④ではペンの使用方法（左ボタンで描画し，右ボタンでスポイトでしたね）で色混ぜができます。ここで注意。スポイトで色を読み取る際にパレットデータをメモリ上に転送します。そのため，色混ぜをしたあとにウィンドウ操作をすると直前に混ぜた部分は保存されません。混ぜ方が気にいらないときにアンドゥ的に使えるのでこのような方式にしてあるわけです。

今月までの機能

次に⑤で左クリックするとペンの色をグラデーションにします。グラデーションは領域内ペイント，カラーチェンジで使用可能です。あと，⑥の位置で左クリックするとその色がペンの色になります。

2)　Lupeウィンドウ

画面上の40×32ドットの範囲を2倍に拡大します。で，上下左右の矢印で範囲を移動します。また，左上方の黄色い部分を左クリックすると拡大範囲が矩形で表示されますから，それを直接マウスで移動させて左クリックをすると，その位置までが拡大範囲となります。描画はペンの使用法と同じです。ただし，1ドット単位の階調なしの描画しか使えません。

4倍，8倍のモードがあってもよかったのですが，拡大範囲が狭くなり，全体とのバランスがとれなくなりますので，敢えてつけませんでした。

3)　Copy

領域指定でコピーしたい範囲を決定して，

図1　パレットウィンドウの様子

自由自在のコピー機能

コピー先で移動してクリックするとグラフィックのコピーができます。コピールーチン自体には任意の1色を透明色とみなしてコピーする機能があるのですが，現段階では封印されています。この機能は「諸設定」ウィンドウの追加で設定できるようになるのでそれまでは我慢してください。

コピー機能を使う際の注意点がひとつあります。閉曲線型で領域指定する際に横幅が39×8ドット以上の指定をすると勝手にキャンセルされてしまいます。これはプログラム上の都合です。

どうしてもコピーしたい場合は矩形で指定してください。

画餅のコピー機能

画餅のコピールーチンは8プレーンすべてにコピーするように設計されています。しかし，それだけではどこにでもあるコピールーチンと変わりません。そこで画餅では任意の1色を透明色とみなしてコピーすることを可能にしてあります。これによって重ね合わせが容易になります。DMACSのようにプレーン別のコピーや複数の透明色を指定するようなコピールーチンもやればできないことはないのですが，画餅ではなるべくハードの機能や制約が見えないようにしているほか，設定方法の簡略化などもあわせて，こういったものはつけないことにしました。

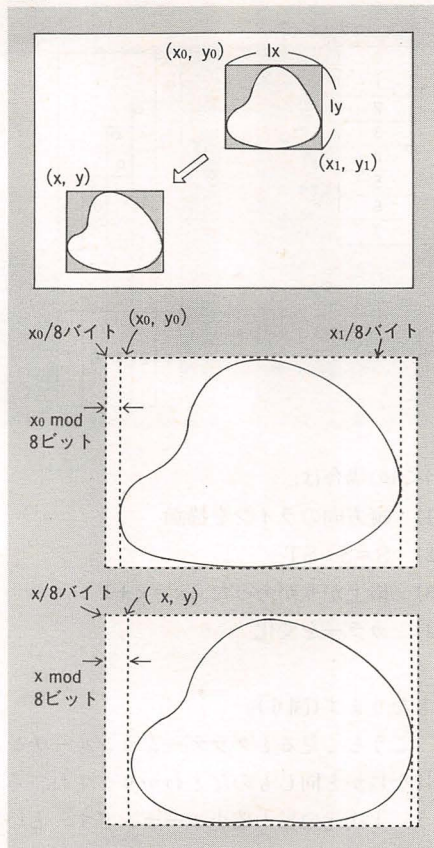
それでは処理の説明をしましょう。このルーチンも水平型VRAMの呪いを受けておりシフトが必要になります。さて，コピー元，コピー先が図2のようになっているとしましょう。すると，処理は次のような手順になります。

1)　まず，1ライン分のデータをG-RAMからワーク上に転送します。透明色を用いる場合にはこの色のデータも転送します。なお，ワークのデータ形式は図3のようになっています。

2)　ワーク上で $(x-x_0)\bmod 8$ ドット分だけ右側へ（$x<x_0$ なら左側へ）シフトを行います。この作業があるためにわざわざデータをワークに転送しなければならないわけです。それにコピー元とコピー先の位置関係によってシフトの回数が変化して処理速度に影響を与えます。

図2　コピーの例

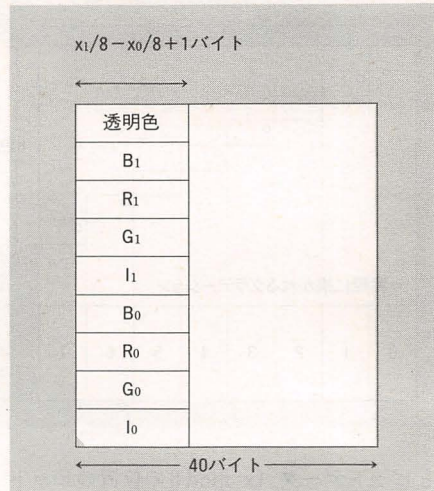
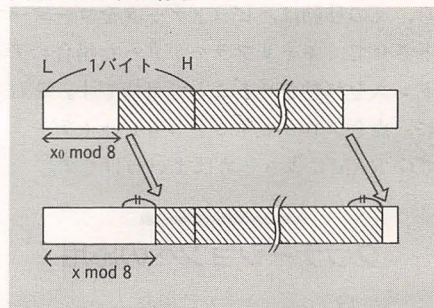
図3　コピー用ワークエリア

図4　シフトの様子

3)　データをG-RAMに転送します。このときには領域データ（透明色を用いるときにはそのデータを含む）でマスクを作り，同時書き込みで転送しています。

4)　以上の操作をlyラインだけ繰り返します。

画餅では裏画面から表画面へデータが移動するためコピー元とコピー先が重なるような場合でも特別な配慮は必要ありません。しかし，もし1画面でコピーを行う場合は位置関係に応じて処理をしなければなりません。コピー先のほうが下にある場合はコピーの順序を下から上へ行います。ここらへんはブロック転送で範囲が重なる場合の処理方法と同じです。

まずはラインから

グラデーションを任意の大きさで実現させようとすると，1色ごとの幅を変化させる必要が生じます。この変化を行う処理を画餅ではラインルーチンを用いることで実現しています。というわけで，グラデーションの処理を説明する前にライン処理を簡単に説明しておきます。より詳しく知りたい方は『試験に出るX1』や1月号のマシン語ゲーム工房，先月号のZバッファの記事などを参照してください。

例として (x_0, y_0) から (x_1, y_1) への直線を考えることにします。ただし，$x_0<x_1$，$y_0<y_1$ それぞれの差をdx，dyとしておきます。

この場合x方向に1移動するときにy方向にdy/dxだけ移動すれば直線が引けるわけです。BASICで書くと，

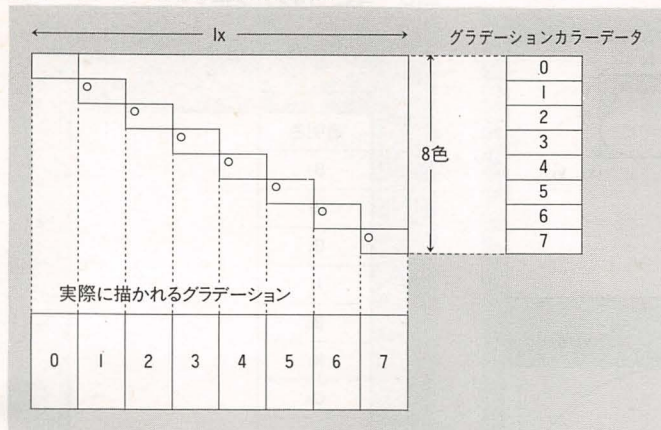
```
ST＝dy/dx：S＝0.5
for x＝0 to dx
    pset(x, y)
    S＝S＋ST：x＝x＋1
    if S≧1 then S＝S−1：y＝y＋1
next
```

となります。ここでSに初期値0.5を代入していますが，この値を変化させると直線を構成する点の位置が若干ずれます。

ちなみに，S＝127/256とするとIOCSのラインと同じ形になります。これをマシン語で実現させるときにはS，STを256倍して処理させます。こうすれば，桁上がりの判定はキャリフラグを見ればわかるのです。

それと座標の位置はG-RAMのアドレス

図5　グラデーションの実際(l≧8)

図6　グラデーションの実際(l<8)

とビットデータ（x mod 8の位置のビットを1にしたもの）で表現できます。これなら，xの移動は，ビットデータをローテートさせて，キャリフラグが立った場合にアドレスを移動させることで可能です。それに，点をプロットするときもビットデータでG-RAMに書き込めばよいわけです。

グラデーションへの応用

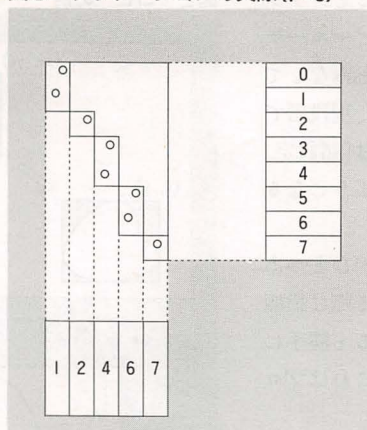
だいたいこんなぐあいでラインルーチンを作ることができます。では，ここでx軸を大きさ，y軸を色数としてみればどうでしょうか？　グラデーションになりそうな気がしませんか？　具体的な例を示せばよくわかると思いますので，横方向のグラデーションの場合を見ていくことにしましょう。

まず，横幅を1とします。そして，画餅は256色8階調ですので色数は8となります。するとラインルーチンでのSTは，

ST＝min(l, 8)/max(l, 8)

となります。SはST－1とすると幅がほぼ均等になります。それで実際の処理はl≧8のとき，

1)　縦方向のラインを描画
2)　S←S＋ST
3)　桁上がりがあったら次のカラーへ
4)　x←x＋1
　　⋮

のようにします。これだけだと理解しづらいと思いますので図で説明しましょう。図5を見てください。Sが桁上がりするときに色を変化させるわけですが，これは○印の位置となります。こうして描画されるとその下の図のようになるわけです。また，

l<8の場合は，

1)　縦方向のラインを描画
2)　S＝S＋ST
3)　桁上がりがあったらx＝x＋1
4)　カラーを変化
　　⋮

となります(図6)。

こうして見るとグラデーションルーチンはなにかと同じものだとわかりますね。そう，1次元の拡大縮小ルーチンです。ということは，データを2次元にして縦と横の両方についてラインルーチンを応用すると……拡大縮小PUTルーチンになるわけです(もちろん若干の修正は必要ですが)。

さらにラインルーチンを組み合わせると，拡大縮小回転PUT，トランスフォームルーチンへと発展していきます。つまり，これらの説明はトランスフォームルーチンへの布石だったわけです。でも，これらのルーチンはもっとも簡単な処理で実現されているにもかかわらず速度は遅いのです（当たり前か)。ですから，ゲームに使ってギャラクシーフォースIIを作ろう！　というのには無理があります（でも MZ-700式の40×25のテキスト画面でやれば似たようなことはやってやれなくはない。と，含みのある発言をする私)。

GCRTCの使い方

画餅の制作初期（約1年前）では，私はG-RAM 4 プレーン同時アクセスの使用法を把握していなかったため1プレーンごとのアクセスをしていました。しかし，この方法で作られたウィンドウ開閉ルーチンなどは遅くてどーしようもないものでした。

そこで試行錯誤で使用法を修得し（解説書には具体的な使用法がなかったので)，G-RAMアクセスの部分のみを変更して画餅のルーチンを作ったわけです。

ということで，ここではGCRTCの説明をすることにしましょう。GCRTCはプレーン同時アクセスのための機能があるのはもちろんのこと，ビューポートの設定，グラフィックモードや表示プレーンの設定，そしてスクロール機能とおいしい機能をたくさんサポートしています。

GCRTCはI/OポートのBCHに内部レジスタ番号を，BDHに与えるデータを出力して動作させます。

G-RAM4プレーン同時アクセス

ここでは画餅で用いているGCRTC設定について解説します。まずは同時アクセス用メモリブロックを論理アドレスに配置します。1ブロックで4プレーン同時アクセスですから256色を使うには2回のアクセスが必要です。これが少々やっかいなのですが，プレーンごとに8回アクセスすることに比べれば楽なものですし，前述のとおり高速です。

画餅では単色用と多色用の設定をして同時アクセスをしています。

・単色モード

まず，レジスタ0～3にFFHを設定します。これらのレジスタは書き込む際のプレーン別のビットマスクとみなせます。そして，レジスタ4に描画するときの色を設定します。これは実際に書き込む場合のプレーンとみなせます。そして最後にレジスタ5に4FHを設定します。この設定でレジスタ4のアクセスするプレーンを指定し，書き込みモードをPSETモードにしています。PSETモードとは，データの1の部分は書き込み，0の部分はそのままというモードのことです。このような設定をしたあとに書き込みの動作に入ります。

また，任意の1色のビットデータを読み込む際にはレジスタ7の第4ビットを1に

下位4ビットをカラー番号にして設定しておきます。そしてG-RAMからデータを読むわけですが，実際に読み込まれるデータはダミーであり，目的のデータはI/OポートBCHから入力します。

・多色モード

いわゆる1バイトデータのGET，PUTです。まず，RGBIプレーンごとのデータをレジスタ0～3に設定します。次にレジスタ4に0FHを設定し，レジスタ6に領域指定でのマスクデータを設定します。PSETモードでマスクデータをG-RAMに書き込むのと結局同じです。しかし，面餅では256色を扱うため2回アクセスが必要だということを忘れてはいけません。すなわち，GCRTCに前もってマスクの設定をしておけば，2回のアクセスの際にZ80側のレジスタのマスクデータを保存しなくてもすみますし，G-RAMに書き込むデータもFFHですみます。このことは保存のために用いていたレジスタが別の用途で使えるということを意味していますので，その分速度面でも有利なのです。

そして，同時書き込みではレジスタ7に0を設定して，G-RAMからデータ（ダミー）を読み込みます。そのあと，I/OポートBCH～BFHからRGBIプレーンそれぞれのデータを入力して実際のデータを得るわけです。

ハードウェアスクロール

スクロール機能はレジスタ0FH～17HにデータRを設定することで使用できます。これだけのレジスタを設定して使うということは単に画面全体をスクロールさせるという以外にいろいろ楽しいことができるということを意味しています。

まずはスクロールエリアの縦のラスタ数をレジスタ16H，17Hに設定することで部分スクロールができるようになります。このスクロールする範囲の上方は非スクロール画面になるわけですが，この画面が開始されている位置をVRAMの好きな位置に指定できます。この設定はレジスタ14H，15Hで行います。

さらにスクロール画面で「このアドレスまできたらVRAMの始めに戻る」という機能のためのアドレスを設定することもでき

図7　特殊なスクロール設定

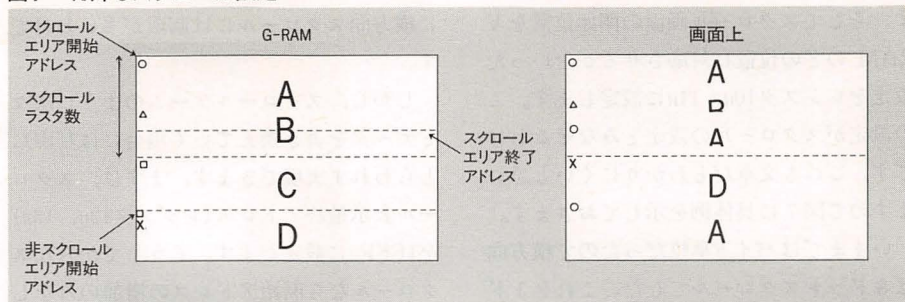



図8　連続スクロール処理

図9　GCRTCのレジスタ

ます。これはレジスタ12H，13Hに設定します。そしてスクロール画面の開始位置をVRAMのどの位置に対応させるかといった設定をレジスタ10H，11Hに設定します。この設定がスクロールの設定とみなせるわけです。これも文章だとわかりにくいと思いますので図7に具体例を示しておきます。

いままではバイト単位だったので横方向は8ドットスクロールでした。これを1ドットスクロールにするにはレジスタ0FHに0～7ドットの設定をします。ただし，ドットスクロールとはいっても所詮は表示開始アドレスとビットをずらしているだけなので横方向スクロールではVRAMの横端で縦方向に1ドットのずれが出てしまいます。また，画面左端に白い部分が出てしまいますので隠す必要が生じます。このように横方向スクロールには制限が多いわけです。

しかし，スクロールゲームのように次々とデータを書き換えていく場合には制限にとらわれず実現できます。まずは，スクロール表示最終アドレス(レジスタ12H，13H)を1FFFHに設定します。そうしてから左スクロールなら開始アドレスの増加のみをします。もちろんそれに対応したビットの移動も行います。G-RAMの書き換えは画面上の左端が都合よく隠してあるので，ここで行うようにします。このときのアドレスは，

(A＋y×40＋x/8)and 1FFFH

を計算した結果にG-RAM開始アドレスを加えることで計算ができるのです（Aはスクロール開始アドレス)。

このようにMZ-2500のGCRTCにはおいしい機能がたくさんあります。そのほかの機能はレジスタの内容がわかれば特に説明せずとも理解できると思いますので図9を参考にして各自でいろいろといじってみてください。

＊　＊　＊

今回は画餅の解説よりもGCRTCの解説のほうがメインのようになってしまいました。次回はグラデーションの説明でほのめかしたトランスフォーム機能が登場します。そのほか文字の大きさ・方向を変更するウィンドウなども登場する予定です。ではお楽しみに。

リスト1 Palette

```
D161 08 00 01 01 0C 01 3F B3 : 09
D169 26 D2 00 01 02 08 03 5C : 62
D171 D2 EC D3 00 02 05 09 07 : A8
D179 26 D3 AD D1 00 0B 02 0C : 90
D181 02 AA D2 32 D2 00 0B 04 : 91
D189 0C 04 C0 D2 42 D4 00 0B : C3
D191 05 0C 05 E0 D2 47 D2 00 : E1
D199 0B 08 0C 08 E5 D2 4C D2 : FC
D1A1 00 0B 06 0C 07 9F D2 17 : AC
D1A9 D4 00 09 02 C9 03 77 05 : 27
D1B1 05 69 49 02 07 04 04 6B : 33
D1B9 4B 02 70 04 04 6A 4A 01 : 7A
D1C1 70 05 0F 69 0F 03 00 08 : 07
D1C9 10 47 1F 02 00 4A 10 56 : 28
D1D1 1F 02 70 08 22 55 45 02 : 57
D1D9 07 57 28 68 40 02 07 57 : 8E
----------------------------------
SUM: 0E 6E B2 AE 27 BA 69 42 D3F6

D1E1 2F 68 47 0B 06 00 07 10 : 06
D1E9 08 01 01 09 24 20 20 20 : 97
D1F1 20 20 20 20 24 24 24 24 : 10
D1F9 00 08 0B 04 09 2A 28 00 : 72
D201 07 00 08 02 01 09 50 61 : CC
D209 6C 65 74 74 65 00 08 0B : 31
D211 02 09 ED 60 00 04 00 80 : DC
D219 10 28 08 18 00 05 00 80 : DD
D221 10 28 08 18 00 3A 61 F9 : EC
D229 CD 32 BF 11 AE D1 C3 3A : 4B
D231 AE FD E5 FD 21 7D D1 3A : 36
D239 FC FA 11 02 01 01 00 00 : 0B
D241 CD A5 BC FD E1 C9 21 39 : 2F
D249 FA 18 03 21 40 FA 4E 21 : DF
D251 00 01 11 0F 05 45 3E 01 : AA
D259 C3 BF BD 3A AC F9 E6 03 : 07
----------------------------------
SUM: ED F5 2E B5 5F 0A 53 8B CAC5

D261 FE 01 C0 CD 8C A2 3A FC : F0
D269 FA F5 CD 5F D3 F1 B7 CA : 60
D271 34 D4 AF CD 97 A4 AF 11 : 7F
D279 01 01 42 4B CD A5 BC 78 : 35
D281 87 87 87 81 5F 16 00 21 : AC
D289 41 FA 19 ED 4B 38 FA 71 : 2F
D291 42 21 01 01 11 05 05 AF : 2F
D299 CD BF BD C3 C8 A4 3A AC : 5E
D2A1 F9 0F D0 CD 8C A2 C3 34 : CA
D2A9 D4 CD BB A4 C0 01 00 00 : C1
D2B1 11 02 01 21 FC FA 7A AE : 53
D2B9 77 CD A5 BC C3 C8 A4 CD : A1
D2C1 BB A4 C0 E5 CD 8C A2 E1 : E0
D2C9 11 01 01 42 4B AF CD A5 : C1
D2D1 BC 79 3C 32 56 FA CD 6A : 2A
```

```
D2D9 D4 CD F3 B9 C3 C8 A4 21 : 9D
----------------------------------
SUM: B5 C2 FD D6 82 35 56 FC BA5D

D2E1 39 FA 18 03 21 40 FA CD : 76
D2E9 BB A4 C0 E5 CD 8C A2 E1 : E0
D2F1 3A FC FA B7 CA 34 D4 3A : F3
D2F9 38 FA 77 DD E5 CD 5F D3 : 6A
D301 AF CD 97 A4 11 0F 05 ED : C9
D309 4B 38 FA 21 00 01 45 7C : 60
D311 CD BF BD CD 66 D3 CD F3 : 0F
D319 B9 DD E1 FD 21 A1 D1 CD : D4
D321 17 D4 C3 C8 A4 3A AE F9 : FB
D329 0F 38 1A 3A AC F9 0F D0 : 1F
D331 CD 8C A2 2A A0 F9 22 69 : 49
D339 F9 2A A2 F9 22 6B F9 CD : 11
D341 83 C5 C3 C8 A4 CD 8C A2 : 72
D349 AF CD 97 A4 3A 61 F9 CD : 18
D351 32 BF 11 1E D2 CD 3A AE : A7
D359 CD 34 D4 C3 C8 A4 AF 32 : E5
----------------------------------
SUM: 03 7C D8 7D BF 87 FD 32 B615

D361 FC FA C3 32 D2 DD E5 21 : A0
D369 D4 0C 3A 39 FA CD 80 BD : 57
D371 21 D7 0C 3A 40 FA CD 80 : C5
D379 BD 21 C0 0C DD 21 D4 0C : 88
D381 06 03 C5 DD 7E 00 77 23 : C3
D389 DD 96 03 0E FF 30 04 ED : A4
D391 44 0E 01 71 23 E5 6F 26 : 61
D399 00 1E 07 54 CD 70 BE E1 : 55
D3A1 72 23 36 7F 23 DD 23 C1 : 2E
D3A9 10 D8 21 3A FA 06 06 C5 : 0E
D3B1 E5 D9 21 00 00 DD 21 C0 : 9D
D3B9 0C 06 03 11 04 00 29 29 : 7C
D3C1 29 DD 7E 02 DD 86 03 DD : C9
D3C9 77 03 DD 7E 00 30 06 DD : E8
D3D1 86 01 DD 77 00 B5 6F DD : DC
D3D9 19 10 E3 E5 D9 D1 0E 00 : A9
----------------------------------
SUM: 87 8E 2F 07 2D 46 A7 87 B4F0

D3E1 DF 5F E1 C1 77 23 10 C7 : 51
D3E9 DD E1 C9 21 41 FA 0E 02 : F3
D3F1 06 08 C5 E5 3E 02 91 57 : E0
D3F9 3E 08 90 5F 4E 06 00 EB : 74
D401 29 29 29 2C 24 11 05 05 : E6
D409 3E 01 CD BF BD E1 C1 23 : 4D
D411 10 E0 0D 20 DB C9 21 39 : 1B
D419 FA 06 08 E5 C5 4E 3E 08 : 46
D421 90 87 67 2E 00 45 11 0F : 11
D429 01 7A CD BF BD C1 E1 23 : 89
D431 10 E9 C9 DD E5 CD 9A C9 : B4
```

```
D439 DD E1 AF 32 56 FA CD 97 : 53
D441 A4 FD 21 AA D1 3A 56 FA : C7
D449 B7 20 1E FD 21 AA D1 21 : AF
D451 03 01 11 0A 0D ED 4B 38 : 9C
D459 FA 06 00 7C CD BF BD B7 : 7C
----------------------------------
SUM: 47 4F 06 3F 89 8B 5C 10 5084

D461 CD A7 BE CD F3 B9 C3 C8 : 36
D469 A4 FD 21 AA D1 21 39 FA : 91
D471 11 03 01 06 0B 3D 28 02 : 8D
D479 06 0E C5 D5 B7 08 78 E6 : CB
D481 01 28 01 23 E5 08 F5 4E : 7D
D489 06 00 3E 01 EB 11 0A 00 : 4B
D491 20 03 11 00 0D CD BF BD : 8A
D499 F1 E1 D1 28 03 14 18 01 : FB
D4A1 1C C1 10 D6 C9 CD 1A BE : 31
D4A9 CD 5A BF 2A 8B F9 11 00 : A5
D4B1 40 19 ED 4B 89 F9 DD 21 : 11
D4B9 39 FA 3A 56 FA 0F 30 36 : 32
D4C1 D9 2A 81 F9 23 CD 66 D5 : A8
D4C9 CD 5A D5 D9 3A 8D F9 1E : B3
D4D1 01 B7 28 05 CB 03 3D 20 : 10
D4D9 FB C5 E5 CD 21 D5 3E 28 : CE
----------------------------------
SUM: A4 EF 1F E3 86 19 84 06 2F11

D4E1 85 6F 3E 00 8C 67 0D 20 : 52
D4E9 F2 CD 41 D5 E1 C1 CB 03 : 45
D4F1 30 E7 23 10 E4 C9 79 D9 : 49
D4F9 6F 26 00 CD 66 D5 CD 5A : C4
D501 D5 D9 C5 E5 CD 18 D5 23 : 35
D509 10 FA CD 41 D5 E1 01 28 : F7
D511 00 09 C1 0D 20 EC C9 3A : E6
D519 5F F9 CD 28 BF 56 18 09 : 83
D521 3A 5F F9 CD 28 BF 7E A3 : 67
D529 57 D9 3E 30 CD 28 BF 79 : CB
D531 D3 BD D9 72 D9 3E 31 CD : F0
D539 28 BF 78 D3 BD D9 72 C9 : 03
D541 D9 19 08 30 0E 08 DA 4F : 69
D549 D5 CD 5A D5 18 F3 CD 5A : 03
D551 D5 D9 C9 08 DC 5A D5 D9 : 63
D559 C9 DD 7E 00 47 0F 0F 0F : 98
----------------------------------
SUM: 32 6E F3 5C 0C 63 40 27 339E

D561 0F 4F DD 23 C9 11 08 00 : 40
D569 B7 ED 52 08 19 08 38 01 : 58
D571 EB F5 CD 70 BE 08 F1 08 : DC
D579 62 6B 2B C9             : C1
----------------------------------
SUM: 13 9C 27 64 A0 21 31 09 9A65
```

リスト2 Lupe

```
D57D 07 00 01 01 0B 01 3F B3 : 07
D585 2D D6 00 02 03 0B 0A 33 : 50
D58D D6 D6 D7 00 01 02 01 02 : 89
D595 59 D6 00 00 00 01 03 01 : 34
D59D 0A 54 D6 00 00 00 0C 03 : 43
D5A5 0C 0A 4A D6 00 00 00 02 : 38
D5AD 02 0B 02 45 D6 00 00 00 : 2A
```

```
D5B5 02 0B 0B 0B 4F D6 00 00 : 48
D5BD 03 77 05 05 69 61 02 07 : 57
D5C5 04 04 6B 63 02 70 04 04 : 50
D5CD 6A 62 03 76 08 10 0F 17 : 83
D5D5 01 70 05 0F 69 0F 02 70 : 6F
D5DD 0F 0F 60 60 02 70 07 17 : 6E
D5E5 67 58 02 70 07 0F 67 60 : 0E
```

```
D5ED 0B 06 00 07 10 08 01 01 : 32
D5F5 09 24 24 20 20 20 20 24 : F5
----------------------------------
SUM: 79 D4 03 0D 49 7C FF 1C 7D6D

D5FD 24 24 24 24 24 00 08 01 : BD
D605 02 09 22 00 08 06 02 09 : 46
```

```
D60D 2C 00 08 06 0B 09 30 00 : 7E
D615 08 01 06 09 32 00 08 0C : 5E
D61D 06 09 2E 00 07 00 08 03 : 4F
D625 01 09 4C 75 70 65 00 00 : A0
D62D 11 BD D5 C3 3A AE 3A AE : 36
D635 F9 0F DA 9A C9 3A AC F9 : 24
D63D 0F D0 CD C3 B7 C3 E0 D6 : 9F
D645 06 01 C3 7F D7 06 02 C3 : EB
D64D 7F D7 06 03 C3 7F D7 06 : 7E
D655 04 C3 7F D7 CD BB A4 C0 : 09
D65D CD 8C A2 DD E5 21 30 FB : 09
D665 11 79 F9 01 08 00 ED B0 : 29
D66D CD 3D A9 AF CD 97 A4 2A : 94
D675 A0 F9 E5 2A 79 F9 22 A0 : DC
-----------------------------------
SUM: 4E B2 BB D8 34 10 70 94 45F6

D67D F9 2A A2 F9 E5 2A 7B F9 : 41
D685 22 A2 F9 CD 98 A3 CD CE : 60
D68D A9 CD D2 A2 2A A4 F9 ED : 9E
D695 5B AA F9 B7 ED 52 28 1A : 36
D69D CD CE A9 2A A0 F9 7D E6 : 6A
D6A5 F8 6F 22 79 F9 2A A2 F9 : C0
D6AD 7D E6 F8 6F 22 7B F9 CD : 2D
D6B5 CE A9 CD C0 A4 28 11 CD : AE
D6BD BB A4 20 CD 2A 79 F9 22 : 0A
D6C5 30 FB 2A 7B F9 22 32 FB : 18
D6CD CD CE A9 E1 22 A2 F9 E1 : C3
D6D5 22 A0 F9 CD B3 A3 DD E1 : 9C
D6DD C3 C8 D7 CD 5A BF CD 25 : 3A
D6E5 D7 C0 3A 38 FA 57 0E 31 : 99
D6ED 06 02 D3 BD 79 C6 02 CD : A6
D6F5 28 BF 73 79 CD 28 BF 73 : FA
-----------------------------------
```

```
SUM: D1 65 39 22 85 6D 2F BC 8527

D6FD 0D 7A 0F 0F 0F 0F 57 10 : 2A
D705 E9 AF CD 97 A4 06 06 3A : E6
D70D A0 F9 A0 6F 3A A2 F9 A0 : 1D
D715 67 11 01 01 3A 38 FA 4F : 35
D71D AF 47 CD BF BD C3 C8 A4 : 6E
D725 DD 6E 01 DD 66 02 FD 5E : EC
D72D 01 FD 56 02 19 7C 26 00 : 11
D735 29 29 EB 2A A0 F9 CB 3C : 07
D73D CB 1D B7 ED 52 ED 5B 30 : 56
D745 FB 19 E5 87 87 47 3A A2 : 2A
D74D F9 CB 3F 90 47 3A 32 FB : 41
D755 80 5F 16 00 D5 EB E6 F8 : 93
D75D 1F 1F 1F 6F 26 00 CD E8 : A7
D765 BE 11 03 FC 19 7E B7 E1 : FD
D76D D1 C0 01 00 40 CD 67 C5 : CB
D775 5F 3A 5D F9 CD 28 BF 7E : 21
-----------------------------------
SUM: FF 98 FD 46 44 F5 5D 48 9453

D77D A3 C9 3A AC F9 E6 01 C8 : FA
D785 C5 CD 8C A2 C1 11 08 00 : 9A
D78D 05 28 24 05 28 0F 05 28 : BA
D795 28 2A 30 FB B7 ED 52 D8 : 4B
D79D 22 30 FB 18 26 2A 30 FB : E0
D7A5 19 EB 21 1F 01 B7 ED 52 : 3B
D7AD D8 ED 53 30 FB 18 14 3A : A9
D7B5 32 FB 93 D8 32 32 FB 18 : 0F
D7BD 0A 3A 32 FB 83 FE B0 D0 : 72
D7C5 32 32 FB AF CD 97 A4 FD : 13
D7CD 21 87 D5 CD D6 D7 C3 C8 : 82
D7D5 A4 CD 60 D8 DD E5 CD 7C : B4
D7DD BF CD 9C BD 01 00 40 09 : 2F
```

```
D7E5 E5 DD E1 2A 32 FB ED 5B : 42
D7ED 30 FB CD E8 BE 09 16 20 : DD
D7F5 1E 05 0E 30 06 02 79 C6 : A8
-----------------------------------
SUM: CD 55 D6 DB E7 75 2C C2 7751

D7FD 02 CD 28 BF 7E 79 CD 28 : A2
D805 BF CD 23 D8 0C 10 EF DD : 6F
D80D 23 DD 23 23 1D 20 E3 01 : 67
D815 23 00 09 01 46 00 DD 09 : 59
D81D 15 20 D5 DD E1 C9 D9 21 : 8B
D825 C0 0C 01 BC 04 ED 78 77 : 69
D82D 0C 23 10 F9 1E 02 21 C0 : 39
D835 0C 16 04 AF 06 04 0E 00 : ED
D83D CB 0E CB 19 1F B1 1F B1 : 5D
D845 10 F4 D3 BD 23 15 20 EB : D7
D84D DD 36 00 FF DD 36 28 FF : 4C
D855 DD 23 1D 20 D9 DD 2B DD : FB
D85D 2B D9 C9 CD C3 B7 DD E5 : D6
D865 FD E5 21 30 FB 06 02 7E : B4
D86D 23 4E 23 CB 39 1F CB 39 : BB
D875 1F CB 39 1F 5A 57 10 EF : F2
-----------------------------------
SUM: F3 0E 62 D8 3F 71 48 6A 5E19

D87D 01 04 03 DD 21 99 F9 DD : 75
D885 73 01 79 83 DD 77 03 DD : A4
D88D 72 02 78 82 DD 77 04 AF : 75
D895 32 C0 0C CD 03 B6 FD E1 : 62
D89D DD E1 C9                : 87
-----------------------------------
SUM: F5 A8 C9 AF DE 3D FD 4A A257
```

リスト3　Copy

```
D8A0 CD 05 B0 C8 CD 10 BC CD : B0
D8A8 91 C0 C8 CD 18 A6 38 F7 : D3
D8B0 2A 85 F9 22 8F F9 3A 8D : 19
D8B8 F9 32 8E F9 CD 3B A6 DA : 3A
D8C0 A2 C0 3E 01 32 64 F9 CD : FD
D8C8 F7 D8 3A 60 F9 CD 2D BF : 1B
D8D0 21 00 60 11 00 80 01 40 : 53
D8D8 1F ED B0 2A 8B F9 22 85 : 11
D8E0 F9 3A A0 F9 E6 07 32 8D : 78
D8E8 F9 2A A0 F9 22 79 F9 2A : 7A
D8F0 A2 F9 22 7B F9 18 C5 3A : 48
D8F8 5E F9 CD 2D BF CD 7C BF : 18
D900 2A 8B F9 11 00 40 19 ED : 05
D908 4B 89 F9 04 ED 5B 8D F9 : 9F
D910 3A A0 F9 E6 07 92 30 03 : 85
D918 C6 08 2B 5F 50 D9 2A 8F : 3A
-----------------------------------
SUM: C1 13 CC 40 FB FF 89 A4 488F

D920 F9 11 00 40 19 3A 53 FA : EA
D928 57 0F 0F 0F 0F E6 0F F6 : 7E
```

```
D930 10 5F 7A E6 0F F6 10 57 : 3B
D938 D9 CD 46 D9 CD B8 D9 CD : F0
D940 DF D9 0D 20 F4 C9 7A D9 : F5
D948 E5 47 3E 07 D3 BC DD 21 : FE
D950 28 78 CD 64 D9 CD 87 D9 : D7
D958 23 DD 23 10 F5 E1 01 28 : 32
D960 00 09 D9 C9 3A 57 FA B7 : ED
D968 28 18 3E 32 CD 28 BF 7B : DF
D970 D3 BD 7E DB BC 4F 3E 33 : 65
D978 CD 28 BF 7A D3 BD 7E DB : 17
D980 BC A1 2F DD 77 D8 C9 D5 : 56
D988 DD E5 C5 0E 32 06 02 AF : 7E
D990 D3 BD 11 A0 00 79 CD 28 : AF
D998 BF 7E DB BC DD 77 00 DB : 03
-----------------------------------
SUM: 3B 88 3E 40 B5 5A 37 D6 8625

D9A0 BD DD 77 28 DB BE DD 77 : 26
D9A8 50 DB BF DD 77 78 DD 19 : AC
D9B0 0C 10 E2 C1 DD E1 D1 C9 : 17
D9B8 7B B7 C8 D5 D9 C1 E5 21 : 6F
```

```
D9C0 00 78 3E 09 08 C5 E5 78 : E9
D9C8 B7 CB 16 23 10 FB E1 47 : EE
D9D0 0D 20 F3 01 28 00 09 C1 : 13
D9D8 08 3D 20 E8 E1 D9 C9 C5 : 95
D9E0 D5 E5 DD 21 28 78 C5 DD : FA
D9E8 E5 3E 06 D3 BC 3A 5F F9 : 4A
D9F0 CD 28 BF 7E DD A6 D8 D3 : 60
D9F8 BD 3E 80 D3 BC 11 A0 00 : BB
DA00 0E 30 06 02 79 CD 28 BF : 73
DA08 DD 7E 00 D3 BD DD 7E 28 : 6E
DA10 D3 BD DD 7E 50 D3 BD DD : A8
DA18 7E 78 D3 BD 36 FF DD 19 : B1
-----------------------------------
SUM: E0 8B 1F 05 62 56 E4 45 5A22

DA20 0C 10 E1 DD E1 DD 23 23 : DE
DA28 C1 10 BB 11 28 00 E1 19 : BF
DA30 D1 C1 C9                : 5B
-----------------------------------
SUM: 9E E1 65 EE 09 DD 04 3C 1B1C
```

リスト4　グラデーション部ソースリスト

```
D4A6:                   8265:           .LPTON
D4A6:                   8266: ;グラデーション--
D4A6:                   8267: ;----------------
D4A6: CD 1A BE          8268: GRADA:  CALL    _MTSET          ;領域マスク:-(領域)and(マスキ
ング)
D4A9: CD 5A BF          8269:         CALL    _GRWMO          ;G-CRTC 単色設定
D4AC:                   8270:         ;
D4AC: 2A 8B F9          8271:         LD      HL,(UTWK+6)     ;左上端 address
D4AF: 11 00 40          8272:         LD      DE,GRAMP0       ;at 4000H
D4B2: 19                8273:         ADD     HL,DE
D4B3: ED 4B 89 F9       8274:         LD      BC,(UTWK+4)     ;(B,C)-lx(bytes),ly(dot)
D4B7: DD 21 39 FA       8275:         LD      IX,GRCOL        ;グラデカラー[0..7]data
D4BB:                   8276:         ;
D4BB: 3A 56 FA          8277:         LD      A,(grmode)      ;グラデモード
D4BE: 0F                8278:         RRCA
D4BF: 30 36             8279:         JR      NC,GRADY        ;縦 gradation
D4C1:                   8280:         ;JR      GRADX          ;横 gradation
D4C1:                   8281: ;X.gradation----
D4C1:                   8282: ;---------------
D4C1: D9                8283: GRADX:  EXX             ;{exx
D4C2: 2A 81 F9          8284:         LD      HL,(dx)
D4C5: 23                8285:         INC     HL              ;lx-(dx+1)
D4C6: CD 66 D5          8286:         CALL    DIVGRD          ;S,ST 設定
D4C9: CD 5A D5          8287:         CALL    GRCST           ;color data セット
D4CC: D9                8288:         EXX             ;}
D4CD: 3A 8D F9          8289:         LD      A,(UTWK+8)      ;-(X mod 8)
D4D0: 1E 01             8290:         LD      E,01H
D4D2: B7                8291:         OR      A
D4D3: 28 05             8292:         JR      Z,GRADX1
D4D5: CB 03             8293: GRADX0: RLC     E               ;ビットデータの作成
D4D7: 3D                8294:         DEC     A               ; (例:A-3 なら 00010000)
D4D8: 20 FB             8295:         JR      NZ,GRADX0
D4DA:                   8296:         ;loop
D4DA: C5                8297: GRADX1: PUSH    BC              ;x,y counter
D4DB: E5                8298:         PUSH    HL              ;GRAM address
D4DC:                   8299: GRADX2: ;draw Y-line
D4DC: CD 21 D5          8300:         CALL    GRSETY          ;dot set
D4DF: 3E 28             8301:         LD      A,40            ;
D4E1: 85                8302:         ADD     A,L
D4E2: 6F                8303:         LD      L,A
D4E3: 3E 00             8304:         LD      A,0
D4E5: 8C                8305:         ADC     A,H
D4E6: 67                8306:         LD      H,A             ;HL:-HL+40
D4E7: 0D                8307:         DEC     C
D4E8: 20 F2             8308:         JR      NZ,GRADX2
D4EA:                   8309:         ;
D4EA: CD 41 D5          8310:         CALL    GRCSET          ;color data セット
D4ED: E1                8311:         POP     HL
D4EE: C1                8312:         POP     BC
D4EF: CB 03             8313:         RLC     E               ;ビットデータ1ドット右へ
D4F1: 30 E7             8314:         JR      NC,GRADX1       ;ビットが右端にきていないと
き
D4F3: 23                8315:         INC     HL
D4F4: 10 E4             8316:         DJNZ    GRADX1
D4F6: C9                8317:         RET
D4F7:                   8318: ;Y.gradation----
D4F7:                   8319: ;---------------
D4F7: 79                8320: GRADY:  LD      A,C             ;ly
D4F8: D9                8321:         EXX             ;{exx
D4F9: 6F                8322:         LD      L,A
D4FA: 26 00             8323:         LD      H,0
D4FC: CD 66 D5          8324:         CALL    DIVGRD          ;S,ST 設定
D4FF: CD 5A D5          8325:         CALL    GRCST           ;color data セット
D502: D9                8326:         EXX             ;}
D503:                   8327:         ;loop
D503: C5                8328: GRADY1: PUSH    BC
D504: E5                8329:         PUSH    HL
D505: CD 18 D5          8330: GRADY2: CALL    GRSETX          ;dot set
D508: 23                8331:         INC     HL
D509: 10 FA             8332:         DJNZ    GRADY2
D50B:                   8333:         ;
D50B: CD 41 D5          8334:         CALL    GRCSET          ;color data セット
D50E: E1                8335:         POP     HL
D50F: 01 28 00          8336:         LD      BC,40
D512: 09                8337:         ADD     HL,BC           ;HL:-HL+40
D513: C1                8338:         POP     BC
D514: 0D                8339:         DEC     C
D515: 20 EC             8340:         JR      NZ,GRADY1
D517: C9                8341:         RET
D518:                   8342: ;dot setXY------
D518:                   8343: ;---------------
D518:                   8344: GRSETX: ;x
D518: 3A 5F F9          8345:         LD      A,(MP2)         ;領域マスクを
D51B: CD 28 BF          8346:         CALL    _SET4H          ;4000h へ配置
D51E: 56                8347:         LD      D,(HL)          ;D-マスクデータ
D51F: 18 09             8348:         JR      GRSET
D521:                   8349: GRSETY: ;y
D521: 3A 5F F9          8350:         LD      A,(MP2)
```

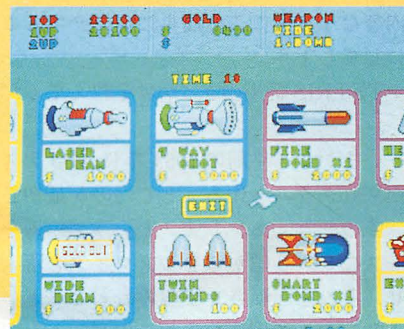
```
D524: CD 28 BF          8351:         CALL    _SET4H
D527: 7E                8352:         LD      A,(HL)
D528: A3                8353:         AND     E               ;ビットデータとANDをとる
D529: 57                8354:         LD      D,A             ;D-マスクデータ
D52A:                   8355:         ;
D52A: D9                8356: GRSET:  EXX             ;{exx
D52B: 3E 30             8357:         LD      A,MPGP0         ;同時アクセスGRAM(bright)を
D52D: CD 28 BF          8358:         CALL    _SET4H          ;4000h へ配置
D530: 79                8359:         LD      A,C             ;C'-カラーデータ(BRGI0)
D531: D3 BD             8360:         OUT     (0BDH),A        ;G-CRTC (レジスタ4)に設定
D533: D9                8361:         EXX             ;}
D534: 72                8362:         LD      (HL),D          ;GRAMへ書き込む
D535: D9                8363:         EXX             ;{exx
D536: 3E 31             8364:         LD      A,MPGP0+1       ;同時アクセスGRAM(dark)を
D538: CD 28 BF          8365:         CALL    _SET4H          ;4000h へ配置
D53B: 78                8366:         LD      A,B             ;B'-カラーデータ(BRGI1)
D53C: D3 BD             8367:         OUT     (0BDH),A        ;G-CRTC (レジスタ4)に設定
D53E: D9                8368:         EXX             ;}
D53F: 72                8369:         LD      (HL),D          ;GRAMへ書き込む
D540: C9                8370:         RET
D541:                   8371: ;gradation colors data set
D541:                   8372: ;----------------
D541: D9                8373: GRCSET: EXX             ;{exx
D542: 19                8374: GRCSE0: ADD     HL,DE           ;S-S+ST
D543: 08                8375:         EX      AF,AF'  ;{af    ;save flag
D544: 30 0E             8376:         JR      NC,GRCSE3
D546:                   8377:         ;L<8
D546: 08                8378:         EX      AF,AF'  ;af}
D547: DA 4F D5          8379:         JP      C,GRCSE2        ;桁上がりなら
D54A: CD 5A D5          8380:         CALL    GRCST
D54D: 18 F3             8381:         JR      GRCSE0
D54F: CD 5A D5          8382: GRCSE2: CALL    GRCST
D552: D9                8383:         EXX             ;}
D553: C9                8384:         RET
D554:                   8385:         ;L>-8
D554: 08                8386: GRCSE3: EX      AF,AF'  ;af}
D555: DC 5A D5          8387:         CALL    C,GRCST         ;桁上がりなら
D558: D9                8388:         EXX             ;}
D559: C9                8389:         RET
D55A:                   8390: ;gradation カラーセット
D55A:                   8391: GRCST:  ;                       ;既に{exx
D55A: DD 7E 00          8392:         LD      A,(IX+0)        ;get color(1byte type) data
D55D: 47                8393:         LD      B,A             ;BRGI0は下位4bit
D55E: 0F                8394:         RRCA
D55F: 0F                8395:         RRCA
D560: 0F                8396:         RRCA
D561: 0F                8397:         RRCA
D562: 4F                8398:         LD      C,A             ;BRGI0は上位4bit
D563: DD 23             8399:         INC     IX              ;カラーデータポインタ+1
D565: C9                8400:         RET
D566:                   8401: ;DE-ST,HL-S in:HL'-
D566:                   8402: ;------------------
D566:                   8403: DIVGRD: ;                       ;既に{exx
D566: 11 08 00          8404:         LD      DE,8            ;8 階調
D569: B7                8405:         OR      A
D56A: ED 52             8406:         SBC     HL,DE           ;Length-8
D56C: 08                8407:         EX      AF,AF'  ;{af
D56D: 19                8408:         ADD     HL,DE
D56E: 08                8409:         EX      AF,AF'  ;af}
D56F: 38 01             8410:         JR      C,DIVGR2        ;Length<8なら
D571: EB                8411:         EX      DE,HL           ;HL-8,DE-Length
D572: F5                8412: DIVGR2: PUSH    AF
D573: CD 70 BE          8413:         CALL    _XDIV           ;DE:-10000H*HL/DE (HL<DE)
D576: 08                8414:         EX      AF,AF'          ;{af
D577: F1                8415:         POP     AF
D578: 08                8416:         EX      AF,AF'          ;af}
D579: 62 6B             8417:         LD      HL,DE           ;DE-ST
D57B: 2B                8418:         DEC     HL              ;S-ST-1
D57C: C9                8419:         RET
D8F7:                   8842:         .LPTON
D8F7:                   8843: COPYWK  EQU     7828H
D8F7:                   8844: ;copy---------
D8F7:                   8845: ;-------------
D8F7: 3A 5E F9          8846: COPY:   LD      A,(MP3)         ;雑用ワークを
D8FA: CD 2D BF          8847:         CALL    _SET6H          ;6000hに配置
D8FD: CD 7C BF          8848:         CALL    _GRWCO          ;G-CRTC 多色に設定
D900:                   8849:         ;
D900: 2A 8B F9          8850:         LD      HL,(UTWK+6)     ;コピー先アドレス
D903: 11 00 40          8851:         LD      DE,GRAMP0       ;4000h
D906: 19                8852:         ADD     HL,DE
D907: ED 4B 89 F9       8853:         LD      BC,(UTWK+4)     ;lx,ly
D90B: 04                8854:         INC     B
D90C: ED 5B 8D F9       8855:         LD      DE,(UTWK+8)     ;D-(x0 mod 8)
D910: 3A A0 F9          8856:         LD      A,(mx)          ;マウスX座標
D913: E6 07             8857:         AND     07H             ;(x mod 8)
D915: 92                8858:         SUB     D
D916: 30 03             8859:         JR      NC,COPY0
D918: C6 08             8860:         ADD     A,8             ;(X mod 8)<(X0 mod 8) なので
D91A: 2B                8861:         DEC     HL              ;アドレス-1 して A-A+8
D91B:                   8862:         ;
D91B: 5F                8863: COPY0:  LD      E,A             ;E-シフト回数
D91C: 50                8864:         LD      D,B             ;save lx
D91D: D9                8865:         EXX             ;{exx
D91E: 2A 8F F9          8866:         LD      HL,(UTWK+10)    ;コピー元: HL
D921: 11 00 40          8867:         LD      DE,GRAMP0
D924: 19                8868:         ADD     HL,DE
D925: 3A 53 FA          8869:         LD      A,(tmcol)       ;透明色
D928: 57                8870:         LD      D,A
D929: 0F                8871:         RRCA
D92A: 0F                8872:         RRCA
D92B: 0F                8873:         RRCA
D92C: 0F                8874:         RRCA
D92D: E6 0F             8875:         AND     0FH
D92F: F6 10             8876:         OR      10H
D931: 5F                8877:         LD      E,A             ;E'-BRGI0 + 10H
D932: 7A                8878:         LD      A,D
D933: E6 0F             8879:         AND     0FH
D935: F6 10             8880:         OR      10H
D937: 57                8881:         LD      D,A             ;D'-BRGI1 + 10H
D938: D9                8882:         EXX             ;}
D939:                   8883:         ;
D939: CD 46 D9          8884: COPY1:  CALL    COPYVD          ;GRAM -> data
D93C: CD B8 D9          8885:         CALL    COPVSL          ;シフト
D93F: CD DF D9          8886:         CALL    COPYDV          ;data -> GRAM
D942: 0D                8887:         DEC     C               ;ly --
D943: 20 F4             8888:         JR      NZ,COPY1
D945: C9                8889:         RET
D946:                   8890: ;copy GRAM->work
D946:                   8891: ;----------------
D946: 7A                8892: COPYVD: LD      A,D
D947: D9                8893:         EXX             ;{exx
D948: E5                8894:         PUSH    HL
D949: 47                8895:         LD      B,A
D94A: 3E 07             8896:         LD      A,07H
D94C: D3 BC             8897:         OUT     (0BCH),A        ;G-CRTCレジスタ7
D94E: DD 21 28 78       8898:         LD      IX,COPYWK       ;ワークポインタ
D952: CD 64 D9          8899: COPYV1: CALL    COPTM           ;透明色と領域データでマスク
D955: CD 87 D9          8900:         CALL    COPCL           ;GRAM -> data
D958: 23                8901:         INC     HL
D959: DD 23             8902:         INC     IX
D95B: 10 F5             8903:         DJNZ    COPYV1
D95D: E1                8904:         POP     HL
D95E: 01 28 00          8905:         LD      BC,40
D961: 09                8906:         ADD     HL,BC           ;HL:-HL+40
D962: D9                8907:         EXX             ;}
D963: C9                8908:         RET
D964:                   8909: ;transparent color
D964:                   8910: ;---------------
D964: 3A 57 FA          8911: COPTM:  LD      A,(tmflag)      ;透明色フラグ
D967: B7                8912:         OR      A
D968: 28 18             8913:         JR      Z,COPTM1
D96A:                   8914:         ;                       ;透明色を用いる
D96A: 3E 32             8915:         LD      A,MPGP1         ;裏GRAM(BRGI0)
D96C: CD 28 BF          8916:         CALL    _SET4H          ;4000hに配置
D96F: 7B                8917:         LD      A,E
D970: D3 BD             8918:         OUT     (0BDH),A        ;BRGI0をG-CRTCに設定
D972: 7E                8919:         LD      A,(HL)          ;dummy
D973: 3E 33             8920:         LD      A,MPGP1+1       ;裏GRAM(BRGI1)
D975: 4F                8921:         LD      C,A             ;C-カラー
D976: 3E 33             8922:         LD      A,MPGP1+1       ;裏GRAM(BRGI1)
D978: CD 28 BF          8923:         CALL    _SET4H          ;4000hに配置
D97B: 7A                8924:         LD      A,D
D97C: D3 BD             8925:         OUT     (0BDH),A        ;BRGI1をG-CRTCに設定
D97E: 7E                8926:         LD      A,(HL)
D97F: DB BC             8927:         IN      A,(0BCH)
D981: A1                8928:         AND     C               ;(BRGI1 and BRGI0)
D982: 2F                8929: COPTM1: CPL                     ;not をとる
D983: DD 77 D8          8930:         LD      (IX-40),A       ;ワークに入れる
D986: C9                8931:         RET
D987:                   8932: ;color > work---
D987:                   8933: ;---------------
D987: D5                8934: COPCL:  PUSH    DE
D988: DD E5             8935:         PUSH    IX
D98A: C5                8936:         PUSH    BC
D98B:                   8937:         ;
D98B: 0E 32             8938:         LD      C,MPGP1         ;裏GRAM(BRGI0)
D98D: 06 02             8939:         LD      B,2
D98F: AF                8940:         XOR     A
D990: D3 BD             8941:         OUT     (0BDH),A        ;BRGI同時読み込みを設定
D992: 11 A0 00          8942:         LD      DE,160          ;40*4
D995: 79                8943: COPCL1: LD      A,C
D996: CD 28 BF          8944:         CALL    _SET4H          ;4000Hに配置
D999: 7E                8945:         LD      A,(HL)          ;dummy
D99A: DB BC             8946:         IN      A,(0BCH)        ;B
D99C: DD 77 00          8947:         LD      (IX+0),A        ,
D99F: DB BD             8948:         IN      A,(0BDH)        ;R
D9A1: DD 77 28          8949:         LD      (IX+40),A
D9A4: DB BE             8950:         IN      A,(0BEH)        ;G
D9A6: DD 77 50          8951:         LD      (IX+80),A
D9A9: DB BF             8952:         IN      A,(0BFH)        ;I
D9AB: DD 77 78          8953:         LD      (IX+120),A
D9AE: DD 19             8954:         ADD     IX,DE           ;HL:-HL+160
D9B0: 0C                8955:         INC     C
D9B1: 10 E2             8956:         DJNZ    COPCL1
D9B3:                   8957:         ;
D9B3: C1                8958:         POP     BC
D9B4: DD E1             8959:         POP     IX
D9B6: D1                8960:         POP     DE
D9B7: C9                8961:         RET
D9B8:                   8962: ;bit shift-------
D9B8:                   8963: ;----------------
D9B8: 7B                8964: COPVSL: LD      A,E             ;bit shift counter
D9B9: B7                8965:         OR      A
D9BA: C8                8966:         RET     Z
D9BB: D5                8967:         PUSH    DE
D9BC: D9                8968:         EXX             ;{exx
D9BD: C1                8969:         POP     BC              ;B-lx,C-bit shift counter
D9BE: E5                8970:         PUSH    HL
D9BF:                   8971:         ;
D9BF: 21 00 78          8972:         LD      HL,COPYWK-40    ;透明色ワークから
D9C2: 3E 09             8973:         LD      A,9             ;9回ループ
D9C4: 08                8974: COPVS1: EX      AF,AF'  ;{af
D9C5: C5                8975:         PUSH    BC
D9C6: E5                8976: COPVS2: PUSH    HL
D9C7: 78                8977:         LD      A,B             ;save lx
D9C8: B7                8978:         OR      A               ;CY-0
D9C9: CB 16             8979: COPVS3: RL      (HL)            ;shift!!!
D9CB: 23                8980:         INC     HL
D9CC: 10 FB             8981:         DJNZ    COPVS3
D9CE: E1                8982:         POP     HL
D9CF: 47                8983:         LD      B,A             ;lx
D9D0: 0D                8984:         DEC     C
D9D1: 20 F3             8985:         JR      NZ,COPVS2
D9D3: 01 28 00          8986:         LD      BC,40
D9D6: 09                8987:         ADD     HL,BC           ;HL:-HL+40
D9D7: C1                8988:         POP     BC
D9D8: 08                8989:         EX      AF,AF'  ;af}
D9D9: 3D                8990:         DEC     A
D9DA: 20 E8             8991:         JR      NZ,COPVS1
D9DC:                   8992:         ;
D9DC: E1                8993:         POP     HL
D9DD: D9                8994:         EXX             ;}
D9DE: C9                8995:         RET
D9DF:                   8996: ;copy work->VRAM
D9DF:                   8997: ;---------------
D9DF: C5                8998: COPYDV: PUSH    BC
D9E0: D5                8999:         PUSH    DE
D9E1: E5                9000:         PUSH    HL
D9E2:                   9001:         ;
D9E2: DD 21 28 78       9002:         LD      IX,COPYWK
D9E6: C5                9003: COPYD1: PUSH    BC
D9E7: DD E5             9004:         PUSH    IX
D9E9: 3E 06             9005:         LD      A,06H
D9EB: D3 BC             9006:         OUT     (0BCH),A        ;G-CRTCマスク
D9ED: 3A 5F F9          9007:         LD      A,(MP2)         ;領域ワーク
D9F0: CD 28 BF          9008:         CALL    _SET4H          ;4000hに配置
D9F3: 7E                9009:         LD      A,(HL)          ;領域ワークデータ
D9F4: DD A6 D8          9010:         AND     (IX-40)         ;透明色とAND
D9F7: D3 BD             9011:         OUT     (0BDH),A        ;G-CRTCに設定
D9F9:                   9012:         ;
D9F9: 3E 80             9013:         LD      A,80H           ;G-CRTC(BRGI)
D9FB: D3 BC             9014:         OUT     (0BCH),A        ; 自動インクリメント
D9FD: 11 A0 00          9015:         LD      DE,160
DA00: 0E 30             9016:         LD      C,MPGP0         ;GRAM(BRGI0)
DA02: 06 02             9017:         LD      B,2
DA04: 79                9018: COPYD2: LD      A,C
DA05: CD 28 BF          9019:         CALL    _SET4H          ;4000hに配置
DA08: DD 7E 00          9020:         LD      A,(IX+0)        ;B
DA0B: D3 BD             9021:         OUT     (0BDH),A
DA0D: DD 7E 28          9022:         LD      A,(IX+40)       ;R
DA10: D3 BD             9023:         OUT     (0BDH),A
DA12: DD 7E 50          9024:         LD      A,(IX+80)       ;G
DA15: D3 BD             9025:         OUT     (0BDH),A
DA17: DD 7E 78          9026:         LD      A,(IX+120)      ;I
DA1A: D3 BD             9027:         OUT     (0BDH),A
DA1C: 36 FF             9028:         LD      (HL),0FFH       ;GRAM書き込み
DA1E: DD 19             9029:         ADD     IX,DE
DA20: 0C                9030:         INC     C
DA21: 10 E1             9031:         DJNZ    COPYD2
DA23:                   9032:         ;
DA23: DD E1             9033:         POP     IX
DA25: DD 23             9034:         INC     IX
DA27: 23                9035:         INC     HL
DA28: C1                9036:         POP     BC
DA29: 10 BB             9037:         DJNZ    COPYD1
DA2B: 11 28 00          9038:         LD      DE,40
DA2E:                   9039:         ;
DA2E: E1                9040:         POP     HL
DA2F: 19                9041:         ADD     HL,DE
DA30: D1                9042:         POP     DE
DA31: C1                9043:         POP     BC
DA32: C9                9044:         RET
```

SOFTWARE INFORMATION

SOFTWARE information

X1/X1turbo
- 闇の壱与(イヨ)伝説・女王塚殺人事件

X68000
- 琉球
- DRAGON
- FORTHクロスコンパイラXMF Z80
- Z80シミュレートデバッガSIM Z80
- スクリーンエディタXe
- TOP給与計算

お馴染みのミュージックにのせて，いよいよ登場のX68000版ファンタジーゾーン。発売はこの7月中旬を予定しているらしいので，もうすでに店頭に並んでいるかもしれません。今度はどんなオマケが付いているかも楽しみですね。

話題のソフトウェア

アフターバーナーやR-TYPEが絶好調のX68000に，今度は電波新聞社から**ファンタジーゾーン**の登場です。お馴染みのミュージックとともに，パステルカラーのファンタジックな世界に展開されるオパオパの大冒険。見ているものをウキウキさせるシューティングゲームの名作に期待していてください。

さて，X1には，今月X68000版をご紹介した**第4のユニット3**がもうすぐ登場してきます。それともうひとつ，秋には光栄から戦国シミュレーション**維新の嵐**が登場します。10月号では恒例のゲーム特集を予定しているので，これらをまとめて，ドーンとご紹介していきたいと思っています。

TETRISの原作者パジトノフ氏来日

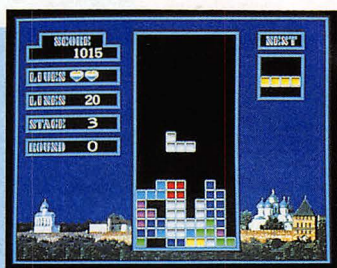
最近，サラリーマンからOLまでと，ゲームセンターで爆発的人気を誇っているTETRIS。このTETRISはソビエトで作られたという話はこれまでにも報じてきたけど，TETRISの生みの親であるゲームデザイナー，**アレクセイ・パジトノフ氏**がビー・ピー・エスの招待により来日され，当社にも

読者が選ぶ今月のゲームベスト10

今月も，予想どおりアフターバーナーが，完全に首位を独走しています。おまけにサイバースティックの発売により，ますますその面白さに拍車がかかっているようです。

一方では，X1版のファンタジアンや信長，そして発売されたばかりの野球道などが順調に票を伸ばし，上位5位のなかに3本も入ってきました。

さらには，待たされながらもようやく発売となったX68000版R-TYPEが，いきなり10位に初登場です。派手さではアフターバーナーには劣るものの，個性的な雰囲気を持つ完成度の高いシューティングとしての評価から考えれば，なかなか侮れないものがあります。さて，これにファンタジーゾーンが加われば，いったいどのような展開を見せてくれるのでしょうか。来月が楽しみですね。

1. アフターバーナー
2. アドバンスト・ファンタジアン
3. スタークルーザー(X68000)
4. 信長の野望・戦国群雄伝
5. 野球道
6. TETRIS(X1/X68000)
7. 今夜も朝までPOWERFULまぁじゃん2 (X68000)
8. リボルティーII
9. Might and Magic II
10. R-TYPE

来日初日，成田空港からいきなり新宿にあるゲームセンターに連れて行かれて非常に驚いたと語る，TETRISのゲームデザイナー・パジトノフ氏(写真左)。中央の写真は，今回氏を招待したビー・ピー・エスのヘング・ロジャーズ社長。もうすでに広告などでお馴染みの顔ですよね。

立ち寄られたので，そのときのインタビュー内容なんぞをここで簡単にご紹介することにしましょう。

Q：TETRIS誕生のきっかけを教えてください。

――いまから5年前の1984年に，IQテスト用プログラムを作ろうとしたのが最初です。そのとき，ペントミノというゲームをベースに5～8個からなるブロックを落下させ，それを回転させて組み合わせるということを考えて作ったのがこのゲームなのです。

Q：日本でTETRISは大ブームとなっていますが，ソビエト国内ではどうなんでしょうか？

――研究所や会社に置かれているコンピュータの，そのほとんどのシステムのなかにこのTETRISが入っているようです。だから，多くの人が職場で遊んでくれているみたいですね（笑）。

Q：現在，TETRISが日本のゲーム界のなかでトップクラスの人気を集めていることについてはどう思いますか？

――みんなが楽しく遊んでくれているというのは，いいゲームだということの証だと思っています。ですから，いまの日本でのブームを見て，そういうゲームを作ったことに対しての満足感を得ることができて，私は非常に嬉しく思っています。

スペースがあまりないので，ここではこれくらいしかご紹介できませんが，このあとの話の続きは今月の「microOdyssey」でご紹介することにしたいと思います。

お知らせコーナー

今月ご紹介しているX68000版**サイクロンExpress**の発売に伴い，アンス・コンサルタンツでは6月14日～7月30日（当日消印有効）までの間，登録ユーザーを対象にバージョンアップサービスを有償で行っています。ご希望の方は，機種名を明記のうえ10,000円を添えて現金書留で下記の住所までお送りください。

〒810 福岡県福岡市中央区平丘町68
㈱アンス・コンサルタンツ
コンピュータシステム部サイクロン係

続いては，**シャープからのお知らせ**です。

C compiler PRO-68Kのプログラマーズマニュアル（499ページ）に誤りがあり，その結果，Cを使って作られた一部のアプリケーションを使用した場合に，S-RAMの内容を書き換えるといった事態が発生しています。それを正常に戻すプログラムを今月の「C調言語講座PRO-68K」（62ページ）のなかで紹介していますので，X68000ユーザーの方は参考までに見ておいてください。

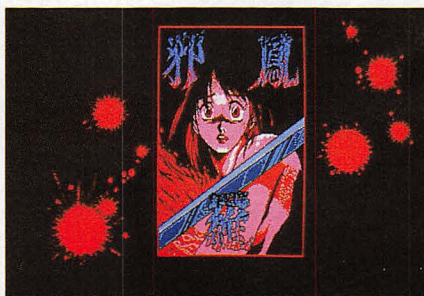
以上，今月のお知らせコーナーでした。

X1/X1turbo用新作ソフト

☆……7月1日現在発売中　★……近日発売予定
*明記されたもの以外の価格については消費税は含まれておりません

★闇の壱与(イヨ)伝説・女王塚殺人事件

ホット・ビィの「Genji」に続くミステリーロマン。今回の題材は，吉野ヶ里発掘で話題にのぼった邪馬台国だ。そこで突然起こった連続少女殺人事件。そのあまりに異常な犯行手口に警察の捜査は行き詰まり，密法に詳しく猟奇的な事件を専門とする探偵に捜査が依頼されることとなった。その探偵が主人公の「不動修羅」である。

捜査が進むうちに次第に見え隠れしてくる新興宗教団体「高天ヶ原の光」の影。そして浮かび上がってくる邪馬台国の後継者，壱与（イヨ）にまつわる伝説。邪馬台国の衰退期，日本史の空白期間に起こった彼女をめぐる争いとは？

襲いかかる妖怪どもを密法でねじ伏せ，修羅は“イヨの復活”を企む謎の集団「高天ヶ原の光」に迫る。

ホラー，ミステリーにアダルト的な趣向も加え，怪奇的なムードで迫る「闇の壱与（イヨ）伝説・女王塚殺人事件」は，タケルソフトで登場だ。

X1turbo用　5″2D版　7,800円
ホット・ビィ　☎03(5261)3900

X68000ソフト&ツールズ

★琉球

このログインソフト「琉球」は，沖縄の風景や民謡をふんだんに盛り込み，沖縄情緒を全面に押し出したパズルゲームだ。といっても，単なる絵合わせではなく，5×5のフィールドに作ったポーカーの役の点数を競うもの。持ち札のなかから好きなカードを選んでフィールドに落とせるが，落としたカードは一番下まで落ちるので，思いどおりの役を作るとなるとなかなか難しい。また，ポイントが各ステージのボーダーラインを超えていなければ，そこでゲームオーバーとなる。

1ステージでの得点を競うTRYモードのほかに，ボーダーラインと見られるカードの数の違う2つの難易度が用意されており，臨機応変な判断力と運がためされる。

もともとは投稿作品だったそうだが，最近のパズルゲームブームの一角を担える強力なゲームといえそうだ。

X68000用　5″2HD版　5,800円
アスキー　☎03(486)7111

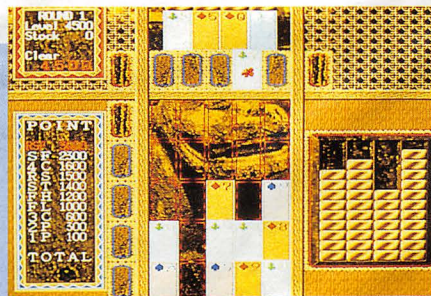
★DRAGON

昨年，MZ-2500用として発売された「DRAGON」のX68000版が登場する。

これは最近流行りのパズルゲームの一種で，現在ゲームセンターでは“四川省”という名前で人気を集めている。このゲームは，古くから中国で遊ばれていたゲームで，戦前に日本に渡ってきたという。中国の一部地域では占いにも使われていたというから，トランプのひとり遊びの中国版といったところだろうか。基本的には麻雀牌を使った神経衰弱のようなゲームだ。

ルールは，2回以上曲がらずにつなげる牌同士を引っくり返して，144枚の牌のすべてを裏返せば完成というもの。このあたりは実際にプレイしたほうがわかりやすいだろう。

一度悩み出すと手も足も出なくなるし，いい加減にやってもやっぱりお手上げになる。とにかく

闇の壱与(イヨ)伝説・女王塚殺人事件

琉球

頭脳プレイに自信のある方にはお勧めの1本かもしれない。特にこのX68000版は，グラフィックや音楽が他機種のものよりグッと強化されての登場だ。

X68000用　5″2HD版　価格未定
ログ　☎03(837)2595

☆FORTHクロスコンパイラXMF Z80/Z80シミュレートデバッガSIM Z80

FORTHは米国のチャールズ・ムーア氏によって考え出された天体望遠鏡観測システムを記述するために考えられた言語だったが，その対話性のよさからシステム記述用言語として，現在では制御システム，コンパイラの開発などに広く使われている。

「XMF Z80」は，CP/M-68KやHuman68k上でZ80用プログラムのクロス開発のために作られた高速3パスのFORTHクロスコンパイラで，64Kバイトをフルに使ったプログラムを作成することができ，また，ソースファイルの分割・リンクやオブジェクトコードの最適化はもちろん，インラインアセンブラを持っているのでクロスアセンブラとしても機能する。

ROM化に適した疑似命令を装備し，RAM/ROMの分割コンパイルが可能。ICEによるシンボリックデバッグのための変換プログラムも付属しており，FORTHによるZ80のクロス開発環境はこのソフトのみにとどまらない。

また，FORTHレベル，アセンブラレベルでのデバッグには，Z80シミュレートシンボリックデバッガ「SIM Z80」も用意されており，「XMF Z80」と同時に発売となった。このソフトにより，Z80カードなどのハードを使用しなくとも，CP/M-80上で行うと同様のデバッグが可能。さらにはOS上でCP/Mで作成されたソフトウェア（ASM，LOADなど）も実行可能となっている。これらのソフトについては，追って詳しく紹介する予定である。

X68000用　各5″2HD版
XMF Z80　58,000円
SIM Z80　50,000円
マイクロフォース　☎03(756)1988

☆スクリーンエディタXe

この「Xe」は，単なるスクリーンエディタというより，全ソースコードが付属したエディタ構造の学習用ソフトである。ディスクのなかにはBASIC，Cで書かれたそれぞれのソースコードが収められており，BASICからCへのコンバート後の効率化の手法などが見ながらに理解することができるようになっている。

また，マニュアルにも高速化についてのメモやユーザーへのメッセージが書かれており，そういった内容を勉強中の人には役立つツールとなるだろう。

エディタとしては，ほぼEDのサブセットというスタンスで，BASICやCOBOLの行番号貼り付け/削除機能なども装備されている。このソフトは，ユーザーにソースを公開するのみならず，そのなかから積極的になにかを学んでもらおうとする試みは，現時点では非常に珍しく，また，価格も手ごろで，今後の動向に期待が持てるジャンルのソフトだ。

X68000用　5″2HD版　4,800円
エル・クラフト　☎0559(71)2015

☆TOP給与計算

シャープブランドの会計用ソフトとして，社員の給与管理を一手にこなす，給与計算ソフトが登場した。このソフトは，会社ごとに設定された基本データを基に，支給控除項目や銀行名，社員ナンバーなどをコード化して登録すれば，給与や賞与の処理，勤怠一覧表の作成から住民税納付書や銀行振込依頼書の発行までを自動的に計算し，処理してくれる。

最大処理人数は2HDのデータディスク1枚につき250人。月給（日給月給），日給，時給および日給＋月給の種別を個人別に設定でき，振込先も端数振込，固定振込，全額振込などの分類が可能。データ入力は明細書のフォーマット上でダイレクトに行え，登録社員の入力・計算状況が表示できるため計算もれの心配はない。給与賞与の明細はそのまま賃金台帳，源泉徴収簿のデータとして保存される。また年度の中途退職者についてもその場で源泉徴収票を発行できる。

勤怠日数は小数点以下まで扱うので有給を半日処理にもでき，勤怠締月を設定すれば有給残の管理も可能。残業時間は分単位で入力可。基準内手当を設定することにより，残業単価を自動算出する機能もある。

DRAGON

TOP給与計算

また，月々の変動項目を登録すればそこだけが入力モードになったり，データ修正後の再計算が自由など，入力面でも省力化が図られているほか，オフィスで使われることを前提に，CZ-8PK3/K6/K8といったシャープ純正プリンタ以外にもPC-PR201F/H/V，HG-2500/VP-1000などにも対応するといった配慮がなされている。そして初心者にはテープの指示に従って使用法を覚えてもらう「トレーニングキット」や，マニュアルのなかに銀行コード・市町村コード一覧も付属しているので，よほど規模の大きな事業体でないかぎり，給与関連の処理を一手に引き受ける即戦力として活躍してくれそうだ。

X68000用　5″2HD版6枚組　200,000円
シャープ　☎03(260)1161

ジョイコントでR-TYPEを遊んじゃえ!

浦「お，さっそくR-TYPEを遊んどるのか」
で「うん。だけどこれ，続けてやってると指先がひきつりそうなんだよね」
浦「またまたぁ。X68000のゲーマーが連射スティックを持っていないはずはないだろうが」
で「だけどねー，このR-TYPEは連射にすると波動砲が撃てなくなっちゃうんだよねー。スイッチをいちいち切り換える暇のあるようなゲームじゃないから，結局は指先がケイレンするまでボタンを叩きつつ速射の連続なんだよね」

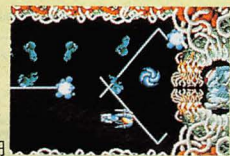
と，いったお悩みをお持ちのゲーマーの方に朗報です。今度，連射トリガーとノーマルトリガーを分離した4ボタンタイプのX68000用ジョイカードが発売されました。その名も「ジョイコントターボ3」。このジョイコントを使えば，R-TYPEの波動砲だって，指を上下に移動するだけで，簡単に使い分けることが可能です。

それでは，先ほどの2人の会話に戻ることにしましょう。

で「わぁ，R-TYPEってこんなに連射できたのか。ほとんどレーザーだぞ，こりゃ」
浦「普通の連射スティックより凄い気がするな。どうだ，耐久力のある敵なんぞは波動砲で一撃じゃぞ！」
で「でえー。なるほど，ボタンが別というのは思いのほか便利だね。特にため撃ちのあるこういうゲームにはとっても便利！」
浦「これで2,200円というのがいいだろう？」
で「うん。あ，話しかけるからゲームオーバーになっちゃった。もう1回」
浦：で「おおっ，こ，これは便利だ！」
浦「ああ，ネームエントリーが」
で「一瞬にしてAAAAAAAになっちゃった」

さあ，どうでしたか。2人ともとっても楽しそうにプレイしていましたね。いまならお買い得価格の2,200円（消費税は別だよ）でご提供しています……。

＊　＊

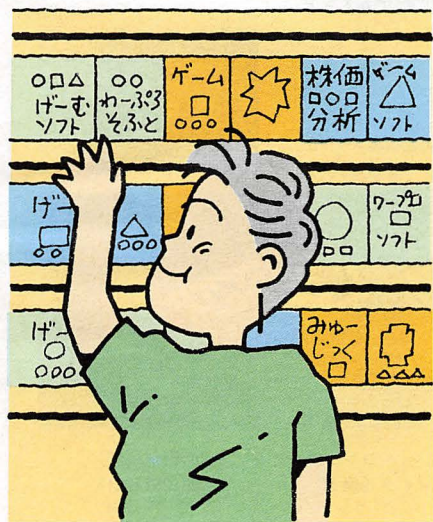
編「なんじゃこりゃ。健康布団のテレホンショッピングか，これは！　まったくいまの若い奴らときたら，締め切りも守らないでテレビばっか見てからに……。でも，この前の電気マッサージ器は安かったよなぁ，ワープロ疲れの肩こりもほぐせそうだし。いまから電話してみよっと……」

X68000版 R-TYPE
アイレム販売7,800円

ジョイコントターボ3　2,200円（税別）
スピタル産業　☎03(251)2918

THE SOFTOUCH

GAME REVIEW

GAME REVIEW

今月は，X1にはなんと，ディスク6枚組のアクションRPG「ガルフストリーム」を，そしてX68000にはぐっと渋く「森田将棋II」と，派手派手バトルアクションゲームの最新作「ジェノサイド」をお届けします。期待してくださいね。

ガルフストリーム

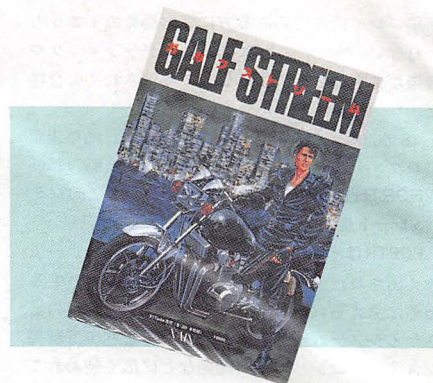
誘拐された恋人を求めて，近未来のダウンタウンをさまよう，アクションRPGです。アニメシーンも豊富に盛り込まれています。

▶いきなり，X1版のアクションRPGである。主人公ボブが，ある女性を助け出すために，広い広い世界をちまちまと歩き回る。情報収集のため，人々から話を聞いたり，ときには敵と戦ったりするのは，いつものパターンである。

デモ2枚を含め，2D版6枚組の力作で，画面の切り替わり方や，会話した際に見られるアニメーション，障害物に半分当たっても勝手に避けてくれる主人公の動き（イースのあの動き），などの細部の技巧からは，制作者の熱意が伝わってくる。また，戦闘シーンは，まるでファミコンのボクシングゲームのような独特のシステムを採用していて，殴られたときの顔も面白い。

しかし，戦闘はやたら難しいし，グラフィックのデッサン狂いも目立つ。BGMもちょっと手慣れてない感じが拭えない。つまりは，全体的に洗練さが不足って感じ。でも，長く遊べることは確かなので，時間のある方にはお勧めかも。

熱中度▶▶▶▷▷▷▷　（お）

▶まったく新しいジャンルのRPGだそうだが，ドラクエのように3人もぞろぞろ連れて歩いたり，イースを思い出させる地下水道には，あまり斬新さは感じない。パッケージは新手の北斗の拳かと思わせてくれるし（実際そんなようなストーリーだ），なによりも，マニュアルのペラペラさはいいにしても，そこに書いてあるイラストには配慮が足りず，あまりにもレベルが低いように思える（絵にはうるさい私）。

随所に見られるアニメーションはいいかもしれないが，ただ，あまりにも随所なので飽きられる可能性もある。いわゆるアニメのON/OFFぐらい付けてもバチは当たらなかっただろう。

戦闘シーンとかは，やたらとお笑い路線狙いなんだけど，これはさすが関西出身と感心してしまう（このソフトハウスは兵庫県）。ばこっと殴ったときの顔とか，仲間のセリフには笑える。でも，やっぱり問題はストーリーなんだよなー。

熱中度▶▶▶▶▷▷▷　（澤）

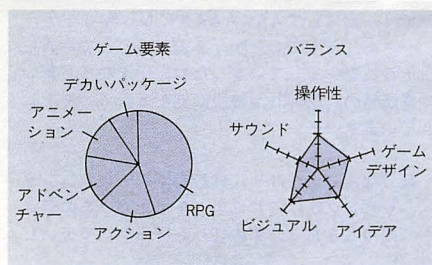
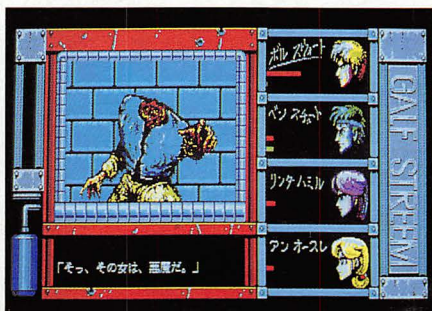
X1turbo用　5"2D版6枚組　8,800円(税別)
（2ドライブ専用）
ザイン・ソフト　☎0794(31)7453

森田将棋II

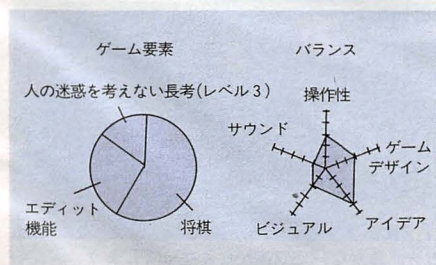
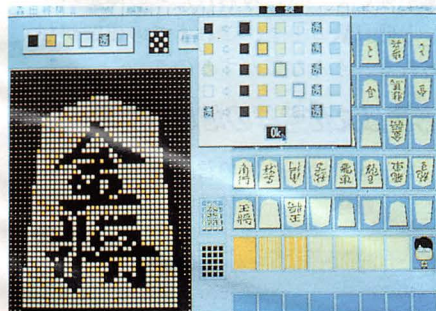
さらに強くなった思考ルーチンに加えて，豊富なエディット機能など，まったく新しくなった森田の将棋の登場です。

▶全国ン千万の将棋ファンの皆様，お待たせしました。ついに出ました X68000版森田将棋II！　いやぁ，将棋ソフトなんて数年前の詰め将棋ぐらいしか知らないもんでね。一気に年寄りになった気分だねこりゃ。マウス対応，駒・盤のエディタが付いてんのはいまや常識。VSのアイコンを書き換えて遊んでるような私みたいな人間には，こいつが結構嬉しい。数種類のデータが用意されてるけど，ツウじゃないと違いがわかんない駒種ばっかなんで，やっぱ自分でエディットするのが正道かな。

相手の王駒を某国の中央委員会主席にしておいて，T-72戦車で固めるときっと燃えるぜ（おいおい）。RS-232Cを使えば北海道のリカちゃんとだって対局できる。

そんでもって忘れちゃならないのが思考ルーチン。これが結構強い。私なんかほと

んど勝てないもんね。こいつはもはや人工知能。え？ お前がヘタなだけだろうって？ うーん，そうかもしんない（笑）。
熱中度▶▶▶▶▷▷▷ （唯）

▶他機種ですでに出ているので，この森田将棋IIというソフトが，「オマケが付いた対局タイプの将棋ソフト」であるのはもう皆さんご存じのとおりなわけで，RS-232Cによる通信将棋モードなどの金のかかるオマケよりも，当然，読者の皆さんのこのレビューへの関心度は，X68000版の「対局モードでのコンピュータの思考ルーチンの強さ」と「思考ルーチンの速さ」に集中してしまうんでしょうね。

私ははっきりいって困っています。というのも，対コンピュータモードはレベルが3つあるのですがレベル1は爽快なくらいテキパキと打っていくんだけど，よほどミスらなければ負けられそうもないくらい弱いし，レベル3では死ぬほど長考だけど，とんでもない強さなのだな，これが。で，中間のレベル2はなんとなく弱い癖に長考だし……。うーん，私は気に入ったソフトではあるんだけど，このタイプのゲームがほかの人の好みに合うかどうかまではちょっとねぇ……。うーん，困った。
熱中度▶▶▶▷▷▷▷ （で）

X68000用 5"2D版 10,000円（税別）
エニックス ☎03(366)4345

ジェノサイド

新鋭ソフトハウスの手による，X68000版オリジナル・アクションゲーム。とにかくこのバトルシーンは必見です。

▶この「ジェノサイド」は，暴走を始めたコンピュータMESIAにたったひとりで立ち向かうバトルアクションゲームなのです。ゲームの内容はというと，次々と現れてくる敵を片っ端から切り捨てていく，なんとなくヴァリスに似た感じの，非常に体力のいるゲームです。

そしてこのゲームの目玉は，なんといっても，やたらとデッカイキャラ同士の高速バトルアクションシーンでしょう。もう，ゲームを始めたとたんに，体中をアドレナリンが駆け巡り，あっという間にプッツンしてしまうほど燃えますよ，絶対に。

ほかにも，背景なんかは滑らかな3重スクロールだし，各面のボスキャラは，先に進むにつれてとめどなくデカくなっていって，最後には6画面分くらいのデカいのが登場したりするんだそうです（まだ見ていないけど）。アクション大好き人間は，完全燃焼できるぞ！ このゲームは。
熱中度▶▶▶▶▶▷▷ （純）

▶そこの人は「SF版源平だね」，あそこの人は「ロボット版ヴァリスだ」とか「サンダーフォースIIだ」という。いずれにしてもこの「ジェノサイド」といい，X68000のオリジナルシューティングは「どうしてこんなにムズカシイんだよぉ」と口を揃えていった。

ムーンサルトをしながら剣を振るう主人公メカ。クセにまみれた敵メカたち。ディスプレイが4台くらい必要な巨大な空飛ぶ要塞。ミサイルを空にバラまきながら長い腕で攻撃してくる工作機械やら，地中から跳び出てくるヘビやら空飛ぶサーフボードに乗ったテッカマンやら，次はどんな敵にまみえるかだけでもワクワクする。

オープニングもセンス良だし，BGMも悪くないし，残すリクエストは“テストプレイし疲れたプレイヤーに合わせて難易度を上げたりしないこと”だけだね。だって，敵がどれも硬いんだもん。まあ，自分だって負けず劣らず硬いけど。
熱中度▶▶▶▶▶▷▷ （K）

X68000用 5"2HD版4枚組 8,800円（税別）
ズーム ☎011(613)0191

次は絶対にハンドルが欲しい！

えーと，うちのX68000にもやっとサイバースティックが届きました。私はまたしてもアフターバーナーにハマッています。

しかしですね，私のいちばん移植してほしいゲームっていうのは，実はチェイスH.Q.とアウトランだったりするんですが，もしチェイスH.Q.がアナログジョイスティック対応で移植されてもあの縦握りの棒と横握りの棒でカーチェイスをやるっていうのもなんか変な気がしてしょうがないと思うんですよね。

はっきりいって，カーゲームやるんなら，ぜーったいにハンドルが欲しい！んで，毎度お馴染みの，ないものねだりのお願いをしてしまうんですが，「シャープさん，お願いです，アナログハンドルコントローラーを作ってやってください！ この私のためにも」アナログジョイスティックの右側にがっぽり被せるタイプかなんかで（シャープの技術に不可能なんかない，ですよね！）。あ，そうそう，ついでにX68000のディスプレイの上に置くパトカーの回転灯もね！

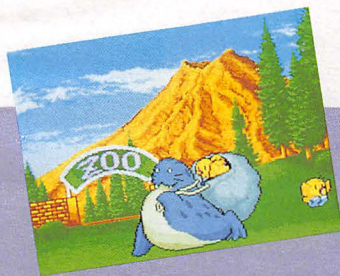
ちなみに，チェイスH.Q.とは，回転灯をつけてぱうぱういいながら走るポルシェのパトカーを，犯罪者の車にがしがしぶつけて遊ぶ，アーケードで流行しているのーてんきなカーゲームのことなんですけど，知ってますよね。 （で）

THE SOFTOUCH

●ニュージーランドストーリー

ティキの不思議な大冒険

Nakamori Akira
中森 章

キウィのティキが捕らわれた仲間を助け出すため，ニュージーランド島を北から南へと大冒険の旅に出ます。敵キャラも登場するアイテムも，どれをとってもかわいいものばかり。でも，難易度はなかなかにシビアな設定となっているようです。

X68000用　5"2HD版2枚組　8,800円(税別)
シャープ　☎03(260)1161

プロローグ

僕はキウィのティキ。キウィという鳥は飛べない鳥の代表のようにいわれているけど，心外だな。それというのも，ここニュージーランドでは天敵がほとんどいないので，いざというときに空を飛んで逃げる必要なんてないからなのさ。翼が退化したってのは神の意志ってわけ。だから，いまのままで僕は十分満足している。いや，満足していたっていうほうがいいのかな。少し前まで，僕たちキウィの仲間はオークランドの自然動物園で楽しく暮らしていたんだ。そのなかにはガールフレンドのピューピューもいて結構ハッピーだったのさ。

でも，ある日のこと，南氷洋から突然やってきたヒョウアザラシのやつにピューピューと仲間たちを連れ去られてしまったんだ。僕は幸運にも逃げ出すことができたけれど，ヒョウアザラシに捕まった仲間たちはニュージーランドの街で売り飛ばされ，辛い思いをしてるらしい。だから，たったひとり残された僕は，仲間を救出する旅に出なければならないんだ。これからしばらくの間，僕の旅に付き合ってくださいね。

オークランドからの旅立ち

オークランドはニュージーランド最大の都市。1840年にイギリスの植民地になって以来，首都がウエリントンに移るまで，25年間首都として栄えてきたニュージーランドの表玄関だ。ここにあるオークランド動物園は，キウィを見ることのできる動物園として有名だという。そういうことに関係があるのかないのか，ともかく，ここオークランドのとある動物園からティキの旅が始まるのだった。

*　　*

まずはライオンさんの檻の前から出発だ。おや，ヤドカリさん，風船に乗ったネコさんやブタさんがいきなり襲いかかってきた。それに原住民(?)はブンブンとブーメランを投げてくる。みんなかわいい顔をしているのに，怖い，怖い。渡る世間は鬼ばかりってやつ。誰だい，キウィに天敵はいないなんていったのは。僕は弓矢で迎え撃つけど，こんなとき空を飛べないのはやっぱり不便だな。それなら敵の風船を奪って飛んでやる。狙いを定めて。ええーい，ジャンプ。ポカッ，ドシャッ。

しめしめ，うまい具合に敵を蹴落として風船を乗っ取ることができたぞ。空を飛ぶっていうのはなんて快適。ふわり，ふわりといい気持ち。この感激を友達のモアくん(19世紀に絶滅した飛べない鳥)にも教えてやりたいな。おや，なんだあの壁の文字は。「Help Me(助けて)」って書いてある。そうか，愛しのピューピューが書き残した文字だ。浮かれていちゃいけない。僕には大事な使命があったんだ。敵の攻撃をかわしながらワニさん，クマさん，キリンさんの檻を通りすぎる。わーい，最初の仲間，見っけ。

青い空，白い雲，緑の森。オークランドは大都市のイメージとはほど遠いのんびりとした街だ。そののどかな風景をバックに空中戦だ。落ちろ，落ちろ，落ちろ。敵を撃ち落とすたび快感になっていく自分が少し怖かったりして。それにしても，さっきから目指す仲間がこの壁の向こうに見えてるのに，そこまで行くには，あそこを通って，ああしてこうして……かなりの回り道。えーっ，制限時間内にあそこまで行くのぉ。

オマケに「Hurry Up」のマークが出てきたら，もう大変。しつこいタイムアップの敵が追いかけてくる。そうでなくても道端のトゲに触れないように注意しなきゃならないし。敵の攻撃は避けなきゃならないし。うーん，なんて難しいゲームなんだ。と，グチをいいつつも何匹かの仲間を助け出してきたけどね（タカヤ！　何事も努力と根性だ)。でも，冷静になってみるとここはやっぱり動物園の前。あのキウィを描いた看板は見覚えがあるぞ。ずいぶん長い間戦ってきたように思えたけど，まだ動物園から外へ出てなかったのか。でも，目の前にいるピンクのクジラさんを倒せばいくらなんでも終わりだろう。なにしろボスキャラだもんね。この手のゲームのパターンだからね。それにしてもピンクのクジラさんは結構手強いよ。氷に覆われたその体がパリーンと砕け散るとき，これまでの死闘がむくわれる。そんな気がするのさ。

お次は文化発祥の地ロトルア

ロトルアは原住民の言葉で「2番目の湖」という意味。ノースアイランドの中央部に

おー，急がないと酸素がなくなってしまう

位置し，温泉と湖の街として有名だ。ヨーロッパ人がニュージーランドを発見する何百年も前に原住民がカヌーに乗ってやってきた土地だという。いわばニュージーランドの文化の発祥地なのだ。２番目の湖の名にふさわしく，ティキが２番目にやってきたのがこのロトルアだった。

＊　＊

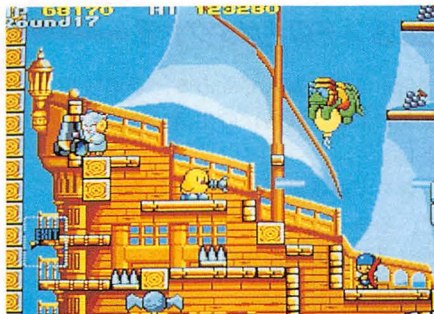
僕が着いたのはロトルアのどこかの村。目の前にはなぜか階段があった。階段の上からはブーメランを持った原住民やヤドカリさんが降ってくる。うへっ，休む暇もないってのはこいつのことだな。僕は空から降ってくる敵をバタバタと倒しながら階段を駆け上っていった。するとそこは村だった。そうか，いままで僕は地下にいたってことか。

しかし，待てよ。村の上には青空が広がっているけど，よく見るとその上にも地面があって村がある。それに上の村には僕の仲間もいるようだ。なんと，ここは階層構造になっている世界だったのだ。もお，P.H.ファーマーもびっくりの世界観。えーっと，上の村へ行くには。そうか，あそこのフロアから上がっていけばいいのか。簡単だね。とは思ったものの，フロアからフロアをジャンプしながら上っている僕に思わぬおじゃま虫。パン，パン，パン。オモチャの兵隊さんが鉄砲を撃ってくる。あぶねーな。兵隊さんが弾を撃ってないときをうまく狙ってジャンプしろってことか。ヒョイ，ヒョイ，ヒョイっと。

ありゃ，上に行きすぎちゃったよ。今度は下に降りなくちゃ。トン，トン，トン。うん，降りるほうは案外楽チンだね。ああ，やっと仲間のところにこられたよ。

ロトルアは湖の街だ。仲間を助けるには湖のなかを進まなきゃならないこともある。そんなとき，僕はピピルマ・ピピルマ・プリリンパと，アダルトタッチでダイバーになりきりっこさ。でも，水中を進むのって結構たいへんなんだ。だって，持ってる武器は水中では役に立たなくなるし，なによりも酸素の心配をしなくちゃならない。酸素ボンベのO₂ゲージが０になると永遠の眠りが待ってるんだ。ナンマンダ，ナンマンダ。

だから，長い間水中にいるときは定期的に酸素を補給しなきゃいけない。初めはこの酸素補給の仕方がわからなくて何度死んだことか。まあ，パパとママの青春を語るいい思い出さ(ナンノコッチャ)。

ところで，ロトルアのボスはタコさんだ。それが，なんとも凄い奴。合体して僕の目の前に現れたかと思うと黒い墨を吹きかけてくる。その墨がコウモリになって襲ってくるのさ。僕はそのコウモリの攻撃を避けながらタコさんの急所の鼻を攻撃し続けてなんとか勝利を得たけど，これも苦しい戦いだったなあ。

仲間を探していざ帆船のなかへ

こいつが大ボス，ヒョウアザラシだ

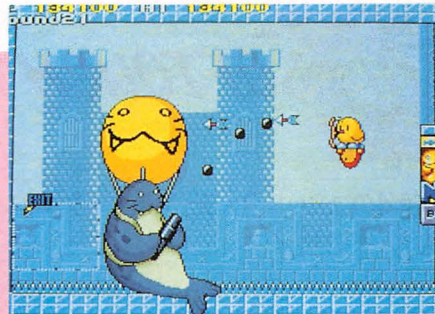
ワイトモ・ケーブはおっきな鍾乳洞

ワイトモ・ケーブは世界８大不思議のひとつである，ツチボタルが生息する有名な鍾乳洞だ。オークランドやロトルアから日帰りで見学することもできる。また，ワイトモ・ケーブ付近は，ツチボタルこそ見ることができないが，数多くの鍾乳洞がある。

＊　＊

墜ちて，いや，落ちていく。気がつくと僕は重力と仲よしになって重力の呼ぶ方向へと向かっていた。つまり落下していた。でも，飛べなくても鳥の仲間さ。地面への着地は大成功の10点満点。こんなことで死んだりはしないよ。あたりを見回すと，どうやらここは洞窟のなからしい。壁にワイトモ・ケーブと書いてあるところを見ると鍾乳洞のなかなのだろう。確かに，同じ地下といってもここはロトルアの地下とはかなり雰囲気が違う。

ロトルアの地下には土の匂いが感じられたけど，ここはただ石という感じだ。なにか冷たさを感じさせる嫌なところだ。早く仲間を助け出してこんなところはおさらばしたいな。僕は例によって敵の風船をぶん捕ったら上へ上へと昇っていく。空を飛んで洞窟に入れば，そこはアミダの迷路。入り口はいくつもあるけど，正しい道はただひとつというやつだな。

どれにしようかな，神様のいうとおり。これだっ。ええいっ。ヒューン(落下する音)。やったね正しい通路だ。僕の勘は世界一さ(共通一次世代は伊達じゃない)。そして迷路を抜けたところは星空の世界。おや，あそこに家があるぞ。こんな洞窟のなかにも街があるのか。ニュージーランドってつくづく不思議なところだね。でも，あの美しい星空がそんな疑問もどうでもいいように思わせてくれる。もしかして，この星空が有名なツチボタルの光なのだろうか(そんなわけないってば)。あるいは飛行石だったりしてね。

いかん，いかん，いまは感傷に浸ってるときではない。仲間のために先を急がねば。走れメロスのようにってね。洞窟を通って，水中を抜けて，空を飛ぶ。これを何度か繰り返すとボスキャラとの対面だ。ワイトモ・ケーブのボスはなんと赤ちゃんをあやすお人形。子供のころはあんな人形であやされたものだ。恐ろしさの前に懐かしさが込み上げてくるよ。うっ，これは敵の作戦なのかもしれない。しかし，敵はおなかからミサイルを撃ってくるだけの単調な攻撃だ。こんな敵にオークランドやロトルアで戦ってきた僕が負けるわけはないのだ。

そうしてサウス・アイランドへ

ティキはクック海峡を渡り，にっくきヒョウアザラシの本拠地マウント・クックのあるサウス・アイランドへやってきた。これからはこれまで以上に苦しいティキの戦いが始まるのだ。(つづく…わけはないよね)

＊　＊

これから，ティキの戦いは佳境に入るが，すべてを話したのではゲームの感激が薄れてしまう。実際にこのゲームをプレイする人のためにこれ以上の説明はやめておこう。これまでの説明でもわかるとおり，このニュージーランドストーリーはかわいいキウィのティキが仲間を助けるためにニュージーランドを転々と旅する物語だ。登場する敵キャラもかわいらしくてほのぼのとした感じを漂わせている。しかしこのゲームは，サンダーフォースIIやドラスピとまではいわないけれど，かなり難しいゲームに仕上がっている。

かわいさだけにつられて買うと，意外にてこずるかもしれないので注意が必要。特に，シューティングゲームやアクションゲーム大好きという人には，絶対おススメ。でも，なんといってもリズミカルなテンポとかわいいキャラたちの魅力が最高だよね。

THE SOFTOUCH

●第4のユニット3・デュアルターゲット

ブロンウィンとタクヤ君の一日

Kunitsu Yoshio
国津 良男

このゲームをプレイすることによってブロンウィン大好き少年となってしまったタクヤ君。でも，このゲームをプレイすると，きっと，みんなそうなってしまうんですよね。では，このゲームの持つ魅力を彼の生活と一緒に覗いてみることにしましょう。

X68000用 5"2HD版3枚組 8,800円(税別)
データウエスト ☎06(968)1236

私の友人その名はタクヤ君

今回は私の友人でもある，予備校生，井森タクヤ君（仮名）の生活についてお話ししましょう。

タクヤ君は最近，妙にそわそわしています。どうしてって，数週間後に，あのコミケ（漫画同人誌即売会）が控えているからなのです。

前回のコミケでは，「みなみりょうこ」さんのサインをもらったとかで，嬉々としていました。それはそれでいいのですが，そのサインを，あたり構わず得意気に見せびらかせるので，周りの人はいい迷惑です。しかし本人は，みんなのそんな心情に，まったく気づいていない様子。むしろ，喜んでくれているものとばかり思っているので手に負えません。

さて，そんなタクヤ君ですが，まだちょっとコミケまで間があるなってことで，暇つぶしにゲームでもすることにしました。本当は受験勉強で忙しいはずなんですが，自分自身のことについては，かなり無関心です。服装を見ればその一端がうかがえるでしょう。タクヤ君は，浪人バッグ（四角いショルダーバッグ）を引っかけ，迷うことなく，秋葉原に出かけました。もちろん親は，彼が予備校に行ったものだと思っています。

タクヤ君は秋葉原に来ると，なぜか深い安堵感を覚えます。駅の階段を下り，馴染みの電気店に急ぎ，X68000用のソフトを物色しました。タクヤ君はゲームを２通りにしかジャンル分けしません。いわく，美少女系ゲームか，それ以外かです。

美少女系のゲームや漫画などは，腐るほど存在してますし，実際タクヤ君も腐るほど食べているのですが，どうも食べた端から腐っていくらしく，不断の補給がないと，生きていけないようです。

で，今回タクヤ君は，パッケージの女の子のイラストに誘われて，「第４のユニット」の第３話，「デュアルターゲット」を選んだのでした。

まずはシステムのご紹介

ここでちょっと，このゲームのシステムについて，お伝えしておきましょう。まっ，いってしまえば，普通のコマンド選択式のアドベンチャーなんですけどね。タクヤ君にいわせれば，美少女系ゲームってことになるのでしょうか。といっても，あの筋のヤツじゃないですよ（「口説き方教えます」は情けないできだった）。

さて，コマンドは，驚く，話す（喜，怒，哀，楽，謝），考える（喜，怒，哀，楽），助ける，戦う（喜，怒，哀），WINDOW（SAVE, LOAD, SOUND），以上６つあります（カッコの内は第２コマンド）。信じられないことに，移動のコマンドがないでしょ，このゲーム。アドベンチャーとはいっても，謎解きとは無縁で，話をしたり考えたりしているうちに，ストーリーはどんどん一方的に進展していくので，移動コマンドなんて必要ないのです。どちらかといえば，シネマウェアと呼んだほうがいいのかもしれません。

それと，元祖「第４のユニット」からの伝統である例の戦闘シーンもしっかりと残っています。今回のシステムでも１対１のみの戦いです。まずブロンウィン，次いで敵が，攻撃方法（ジャンプ，パンチ，キック，ブラスト，シールド）と攻撃位置（頭，右腕，左腕，右足，左足，胸，腹）を決めます。で，あとは画面下方で戦いが行われるのを見るだけ。どちらかが倒れるまで，これを繰り返します。敵は必ず弱点を持っているので，そこを早く見つけ，情け容赦なく集中攻撃をかけるのが攻撃の基本です。

さて，オープニングを見終わって，事情が飲み込めたタクヤ君はいよいよゲームをスタートさせたようです。彼のX68000を覗いてみることにしましょう。

第1章　A BEGINNING

ここは統合軍極東地区作戦本部，つまりはブロンウィンたちの基地です。彼女は自

結局はこの青いBSに博士を連れ去られてしまう

学校に行くとなんとブロンウィンが２人いた

室で，なにやら物憂げな表情をして鏡を見つめています。「WWWF基地のなかで見たもうひとりの私」3カ月前に敵基地内で見た自分とウリふたつのBS（バイオニックソルジャー）のことを考えているようです。世界中には自分とそっくりな顔の人が3人はいるっていいますが，この場合はそんなことじゃなくって，WWWFが作っているクローン人間のことを考えているのです。

と，いきなりエマージェンシーコール。「敵アンドロイドが基地内に侵入し，PPB（サイコパワーブースター，BS用パワーアップ装置）を奪い逃走中！」。それを聞いて部屋を飛び出てみると，案の定アンドロイドが3体，目の前に立ちふさがりました。PPBと，その開発者の越中博士が彼らの腕に抱きかかえられています。仲間のアッシュも応援に駆けつけました。ブロンウィンは青いアンドロイドに的をしぼって戦闘を仕掛けます。ここで画面は変わって，バトルシーンとなります。

ブロンウィンはまず，胸部パンチを繰り出しました。アンドロイドは腹部キックで責めてきました。結果は？　体力の減り具合を見てみると，胸部パンチは結構効いたようです。ブロンウィンもそれなりのダメージを受けましたが，敵ほどではありません。このあとブロンウィンは，胸部パンチ一本槍の攻撃を続け，9ターン目で，なんとかアンドロイドに勝利したのです。

画面は戻って，残る2体のアンドロイドが進退極まっています。この調子で，全滅させようかと思っていると，サイコブラスト（カメハメ波みたいなもの）が上方から飛んできました。なんだ，と思って見てみると，青いプロテクターで身を包んだ敵BSが崖の上に立っています。その隙をついて2体のアンドロイドは逃げ出し，青いBSとともに去って行ってしまいました。結局，PPBも博士も奪われてしまったのです。

第2章　THE DESTRUCTION

基地内本部で，ブロンウィンとアッシュ，セスの3人が集まっています。3人とも17歳の女の子のBSです。アッシュとセスは博士の救出に行き，ブロンウィンは奇襲に備え，待機することになりました。

よって後方待機と決まったブロンウィンは，ひとまず学校（私立星雲学園）に遅刻しながらも出かけることにしたのです。その教室で友人たちと会話に興じています。ところがなにかブロンウィンの様子がおかしい。そのうち，気でも狂ったのか，サイコブラストを四方八方に放出し始め，クラスメイトは大騒ぎ。

病床のブロンウィンを気づかう仲間たち

と，ところが，どういうことなのでしょう。そこに，ブロンウィンがもうひとり，教室に駆け込んで来たのです。そう，こちらが本物のブロンウィンだったのです。

狂ったかに見えたブロンウィンは，WWWFのBS，すなわちブロンウィンが以前，敵基地内で見たそっくりさんでした。その彼女が名乗りました。「なにを驚いている？　BSはクローン兵器だ。ブロンウィンはお前だけではない！　私の名はスイシーゼ。お前の力，試させてもらう！」。そんなこんなで，再びバトルシーンへと移るのでした。

ブロンウィンは少々手こずりながらも，スイシーゼに勝つことができました。頭へのパンチが効いたようです。しかし，スイシーゼは最後の力をしぼって，何発ものサイコブラストを放ったから，さあ，たいへん。そのうちのひとつが，女生徒に向かってまっしぐら。

危ない！　そう思ったブロンウィンはとっさに飛び出し，身を挺してその女生徒をかばいました。

しかし，この行動に驚いたのはスイシーゼです。彼女には，人を助けるというブロンウィンのこの行動が不可解だったからです。「な，なにぃ？　自分から飛び出さなければ，余計な傷を負わずにすんだものを」。ブロンウィンは答えます。「あなたたち，WWWFの人間が知らないことよ。これが人を助けるということなのよ！」。「助ける？　他人の代わりに傷ついてなんになる！」。しかし，その言葉とは裏腹に，スイシーゼは動揺を隠せません。

そこに，背後から現れたのが例の青いBS。「どうしたNo.7（スイシーゼ），No.4（ブロンウィン）を殺せ」

そしてタクヤ君は夢の中へ

ここまでで1時間少々。タクヤ君は昼間歩き回った疲れが出てきたためか，セーブをして，X68000の電源を落としました。いまは2章の途中です。これから，第3章「THE RESURRECTION」，第4章「A POWER」と続きますが，この調子でいけば，明日か明後日には順調に終わることでしょう。戦闘で負けさえしなければ，行き詰まることはないのです。といっても，エンディングは3通りあるようなので（私は2通りしか見ていないけど），一度や二度はやり直す必要が出てくるかもしれません。とにかくこの3作目は，マウス対応になっているから進行がスムーズに流れるようになったほか，グラフィックやサンプリングも気合が入ってるようだし，これまで以上の楽しみ方ができるのは請け合いのようです。

スイシーゼが心を開く感動のシーン

そのほかにも「SAVE」コマンドを選ぶと瞬時にそのシーンをセーブしてくれるし，「待て」だとか「？（入力待ち）」，「PUSH（文章の続きがある場合）」なんかのお知らせアイコンも用意されているしで，結構サービス精神も満点です。ただ「喜怒哀楽謝」のコマンドが，プレイしていてちょっと感覚的にわかりづらいのが気になったのですが……。

＊　　　＊

タクヤ君は床に就き，目を閉じ，いまのゲームのことについて，いろいろと思いを巡らせています。

「えーとブロンウィン，それからブロンウィン，そんでもってブロンウィン……」

ちょっとぉ，ほかのことも考えてよ。このゲームって音楽にはあまり力が入ってなかったようだけど，タクヤ君はどう思う？

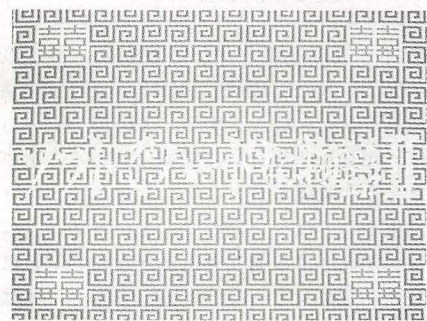
「うーん，ブロンウィン……」

ストーリーはシリアスなんだけど，なんとなくノリが軽すぎる感じだよね。

「やっぱり，ブロンウィン……」

と，タクヤ君は一向に，視野を一点から広げない様子。そうして，ブロンウィンの残像を瞼の裏に残したまま，満足気な笑みを浮かべながら，スヤスヤと深い眠りへとおちてゆくのでした……，とさ。ちゃんちゃん。

THE SOFTOUCH

●ソフトでハードな物語2

泣くに泣けない業界残酷物語

Komura Satoshi
古村　聡

続編なんだから，前作の調子でのほほんとプレイしていれば丸く収まるものだと思っていると，そう簡単にはいかないのがこの世の中。特に扱っているネタがネタだけに，常識では考えられないような落とし穴があったようですよ，このゲームは。

X68000用　5"2HD版 4 枚組　7,800円(税別)
システムサコム　☎03(635)7609

まっとうな人生を歩みたい人にとっては，踏み込んではならないふたつの世界がこの世にはあるという。ひとつはいわずと知れたパソコン雑誌の編集部。そして，もうひとつがこれから皆さんをご招待するソフトハウスであったりする。

そう，この物語は危険なその筋に片足を突っ込んでしまった青年への鎮魂歌（レクイエム）なのである。もうあの日には帰れない……。

後悔の嵐

「うーん」，締め切り前夜。まあ，こんなことになるんじゃないかとは思ってはいたんだけど，やっぱりなってしまった。目の前のワープロ文書は真っ白。いや，背景は黒だから真っ黒か。要するに文章が全然書けてない。だいたい，あのとき簡単に安請け合いしてしまったのがいけなかったんだよな，きっと……。

＊　　＊

「おまえ，シナリオに青春を賭けてみないか？」

モカシステムの社長である親父にそういわれたとき，すごーくいやな予感がした。だいたい，こういうことに限って唐突にやってくるんだもん，なんか呪われてるのじゃないかとさえ思ってしまう。あのときもそうだった。ちょうど１年前，僕がいきなり親父の会社の社長をやらされたときだ。うちの親父ときたら年も考えずにむちゃくちゃやるから，過労でばったり倒れて病院にかつぎ込まれて，

「私の療養中，私の代わりに社長をやっててくれー」

だもんな。んで，親父の会社がソフトハウスだったもんでゲームソフトの開発をやったんだけどそりゃもう，大騒ぎ。うーん，思い出したくない。

で，今回は親父の会社で「アドベンチャーゲーム短編集」を作るんで，そのうちの１作の原作・シナリオを担当することになったわけだけど，いきなり「賭けてみないか？」だもんなー。まぁ，ご機嫌をそこねちゃ悪いから「はいはい」ってふたつ返事で聞いちゃったけどさ。ほんとは，てめえいいかげんにしろよな，っていう感じ。

いや，それ以前にやっぱりあの辺からまずったのかな。学校で学祭のあったあの日，

「学祭見に行くのもつまらないし，いいや，ひまだから久しぶりにモカシステムにでも遊びに行っちゃえ」

なんて考えたのが運のつきだったのかな。久しぶりにみんなに会えたのは嬉しかったけど，まさかこんなことになるとはなー。それに，最悪なことにあのあと沙織さんを泣かせちゃったし……。ちょーっとやばかったかなぁ。

ま，それはそれとしてだよ。満を持してこの原稿の手書きプロットを持っていったときのチーフの堀田さんのあのセリフ！

「うーん，いまいちだなぁ。ま，素人さんだからしょうがないか」

くやしいったらありゃしない。

そういえば，会社に新しく入った石田さん，ありゃいったいなんなんだろう？　履歴書には家族なし，経歴なし。なぞのリターンオブイシダ。なーんて，あーっ！　ひとりでウケないギャグいってどうすんだ。変な回想シーンやってたから，さっきから全然原稿が埋まってないじゃないかーっ！

(Fade Out)

ねぇ，なぜなの!?

あ，おはようございます。やぁ，一昨日の夜は本当にまいりましたけどね。なんだかんだでぜーんぶ，昨日のうちに原稿アップしてもう堀田さんに渡しちゃってるんだもんね。あとは，今日からモカシステムのアドベンチャーツールでデータを作ってやるだけになるはず。はは，楽勝楽勝。さーて，ほんじゃ行ってきますかね。

「どーもーっ！　宏志でーす」

「あ，宏志君……」

ありゃ!?　モカシステムに着いたはいいが，チーフの堀田さんの顔が妙に暗い。

「どうしたんです？」

「あのね，君のプロット，昨日の会議で評判悪くてね，悪いけど今回は見送らせてもらうことにしたよ」

「ボツってことですか？」

「ありていにいえばそういうこと」

そして，堀田さんの「次回に期待してるよ」の台詞を背に，僕はトボトボとモカシステムをあとにした。

＊　　＊

某月某日，編集部で「ソフトでハードな

石田さんって仕事はできるんだけどなんかヘン

物語2のレビューやらない？」といわれて「やるー！」とあっさり返事をしてしまった私がきっと悪いんだろう。それとも前作と同じようなものだと考えていた私が悪いんだろうか？　どっちにしても私が悪いのに違いない。でも，信じられなかった。前作ではよっぽど自分から失敗しようとしなければ，絶対ゲームオーバーにならなかった。しかしなぜか，私はその2作目を前にして悪戦苦闘している。

えーい，しょうがない，困ったときの友だち攻撃だ。

「あのさぁソフトでハード2でさ，つっかえるんだけど，なにが悪いのかなあ」

私はPC-98(っていうより，マシンはPC-286なんだけど)版を終わらせた友人に聞いてみた。

「あ，それねぇ，堀田さんとか小田さんとかメインになる人物に必ず話しかけておかないとシナリオ，ボツにされちゃうんだよね」

そういうことだったのか。よしよし，これでなんとかなるだろう。

えっ，それでもだめ!?

あ，おはようございまーす。さーて，はんじゃ行ってきますかね。

「どーもーっ！　宏志でーす！」

「よう，ちゃんと朝9時にきたね。感心，感心。それじゃ，10時から企画会議やるから会議室に集合ね」

「はい，わかりました」(あー，よかった。やっと進んだ)

それから，お昼ごろになって会議が終わった。でも，なんかヘンなんだよな。なにがヘンかって，まずチーフの堀田さん。みょーにあ・かるい。しかも普通のあ・かるいんじゃなくて，ざーとらしくあ・かるい。んで，ほかのみんな（これは朝来てから思ってたんだけど）もなんかおかしい。朝の9時から誰ひとりとして遅刻しないで黙々と働いてる。そう，おしゃべりする声さえ聞こえず，音もたてず黙々と。こう書いていくとなんか普通の会社っていうのはそれでいいんじゃないって思うかもしれないけど，少なくともパソコン関係の会社だとしたらこの雰囲気は絶対おかしい。だいたい，たとえばOh！X編集部なんかの場合は，毎日ほとんど小学校の遠足バスみたいにうるさいものだ。これはぜったいおかしい……。

そのあと，マック氏や女の子たちに聞いてみたところどうも，原因には2つあるみたい。まずひとつは一部のスタッフが会社に不満を持ったり辞めたがったりしていること。もうひとつは新入りの石田さんが女の子に異常にウケが悪いみたい。確かに，スタッフに時間厳守させたり，仕事の指図をしたりと新入りの態度ではない。彼女は生意気と受け取られてもしょうがないタイプみたいだし。

などといろいろ考えていたら，ついに受け付け嬢のヒロポンと石田さんが喧嘩を始めた。と，いうよりヒロポンが石田さんのあまりに非常識な態度（なんでもヒロポンが手が空いたからと雑誌を読んでいたら，石田さんが「仕事中，ふざけたことをするな」といったんだそうだ）が頭にきたのか，ぷっつんきれてしまった。周りのパソコンなんかに当たり散らすヒロポン。まずい，止めに入んなくちゃ！

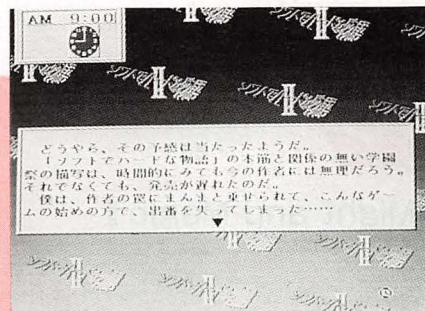
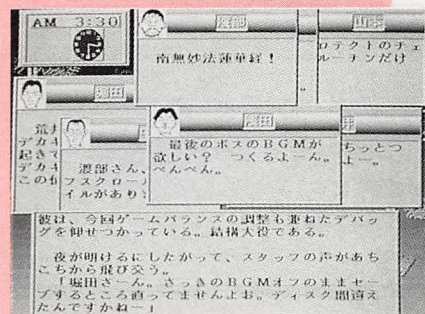
ヒロポンの件が一段落したので，ちょっと気休めに話でもしてみる。音楽&シナリオスタッフの浅田君に会社の雰囲気のことを聞いてみた。すると，浅田君は同業他社がスタッフに誘いをかけているらしいこと，そして，浅田君自身もいま手掛けている新作が終わったらモカを去るつもりであるという。そのあと，小田さんから中島氏裏切り説，アメリカのソフト会社の日本上陸作戦の噂も聞いた。

どんどん，悪くなる会社の雰囲気，気まずくなる人間関係，裏切り，スパイ，外からの圧力，そして迫ってくる新作の発売予定日（3月）。

2月末，なんとか僕はギリギリの日程でシナリオを仕上げた。しかし，僕以外のスタッフは全然シナリオが進んでおらず，3月発売は絶望的になった。そして，やはり恐れていたことが起こった。スタッフやプログラマが集団で辞表を提出し，完全にソフトの開発が不可能になった。ソフトが出なければ収入は当然ない。そして，ついに残ったスタッフの給料も払えなくなる。そうしてついに，モカシステムは倒産してしまったのだ。

再び始めたある日のこと。僕は新入りの石田さんが横浜でヘンな外人と楽しそうに話ているのを見た。確かモカシステムで聞いた話では，石田さんはこの何日も無断欠勤が続いていたはず。おかしい。もっとも，こっちも美奈子と一緒だったので尾行するわけにもいかなかったのだが……。そして，僕が帰ってきたときモカシステムは蜂の巣をつついたような大騒ぎになっていた。

＊　　　＊

えっ？「これは本当のエンディングではありません」だって。えーっ！　まだ真のエンディングじゃないのーっ！　冗談じゃ

なーんにも考えないでプレイするとこうなる

マスターアップ前夜はまるで戦場

ないよ。もう，原稿を書きながらも，もう42時間ぶっ通しの徹夜でこのゲームやってんだぞ。しょうがない，また友人作戦，だぞっと。

「ねー，ねー，ソフトでハード2の真実のエンディングへの進め方って知らない？」

「へっ？　ソフトでハード2？　あれって，みんなに気を使って誰にも話しかけるようにすればいいだけでしょ，あんなもん」

「えー，だって終わらないよー」

「んな，ばかな」

「だって，もう徹夜してやりまくってるんだぜ。間違うわけないでしょ」

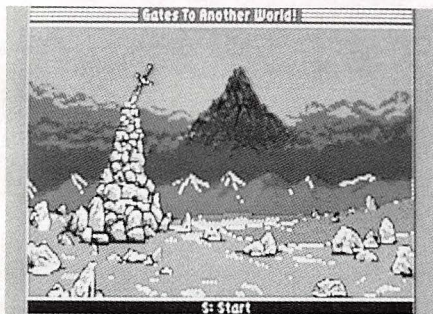
「えー!?　そんなはずは，ぜっーたいにねぇよ」

はっきりいって，もう私には解けそうもない。仕方がないので編集部に電話を入れた。

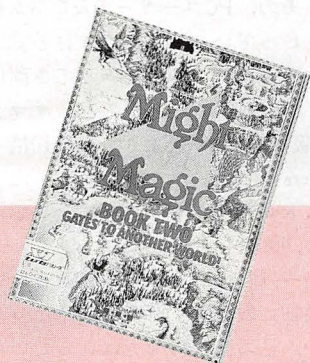
「もしもし，(て)ですけど。あのー，ハッピーエンドで終わる方法ってわかりませんか？　あの，PCユーザーの友だちがなんかおかしいっていってるんですけど」

「あ，それね。このゲームってさ前作より10倍くらい難しくなっててね，オマケにX68000版はPC-98版よりさらに10倍くらい難しいんだって。ソフトハウスさんの話だと，ストレートにハッピーエンドで終われるのは，500人にひとりくらいだろうってさ」

あのー，簡単にいうけど，普通そういう話って最初に教えてくれるべき話だったんじゃゃ……，私の徹夜の努力はいったいなんだったつーの！　でぇー，レクイエムじゃ，レクイエム。ついでにバームクーヘン喰って寝てやる。

●Might and Magic Ⅱ

RPGのプロ養成講座 上級編

Shimizu Kazuto
清水 和人

3回に渡ってお届けする，この「RPGのプロ養成講座」もようやくクライマックスへと突入します。マッピングのコツから，効率的なアイテムの使い方や戦闘方法までと，ゲームのヒントを交えて進んでいきます。それではじっくりお楽しみください。

X1/X1turbo各専用5"2D版5枚組9,800円(税別)
(2ドライブ専用)
スタークラフト ☎03(988)2988

都会の夜は，窓を開けると遠くに車の行き交う音が聞こえ，時折バイクのエンジンを吹かす音や，アスファルトの道を歩く人の足音をアクセントにして静かに更けてゆく。こだまするような遠くの電車の音も午前1時を過ぎると聞こえなくなり，大巨人東京は2～3時間の軽い睡眠をとる。最も静かになる3時頃，私はふと目を覚ました。

画面はA-3エリア(7, 7)，グルメのとっつぁんのキッチンで不覚にも寝てしまったのだ。ついさっきこの近くでドラゴンの奇襲を受けた疲れが出たのだろうか。ツンドラの町のラッキードッグサロンで「Roast Leg of Wyvern」を連夜たらふく食べたせいだろうか。X1のキーボードを抱えて眠ってしまったのだ。首を回してこりをほぐし，そろそろ夜の戦いに入ろう。

外は妙に静かだが，部屋のなかではM&Mのアタック音が「バーン，バーン」と響いている。これこれ，このノリですよ。人生こうでなくっちゃあ。

マップこそ命

M&Mの付録のマップ。どこかへやってしまったなんて人はいないと思うが，これは非常に貴重なヒント集になっている。各エリアのどのへんが山で，どこが森や川なのかがひと目でわかる。もちろん，川や湖や海は僧侶のレベル3-6の呪文「Walk on Water」で，山のなかはミドルゲートで2人以上が身につけた「Mountaineer」のスキルで，森ではやはり2人以上がミドルゲートで身につけた「Pathfinder」のスキルで通れることぐらいわかっていてほしい。

そのほか，各町や城の位置，そして危険なツンドラや砂漠の場所もしっかり描かれている。戸外を歩くときは必ずマップを見ていれば，次になにが起こるかだいたい見当がつくようになっているのだ。さらに，その場所にいる特別強い敵も描かれている。これらは，クエストに関係する場合が多いので，このマップがあれば次にどうしようかということが簡単にわかってしまうのだ。それだけこのマップは力作なのだ。また，画面に出てくるグラフィックより遥かに変化に富んでおり，冒険もひときわ興味深いものとなろう。

強くなってからの戦い方

レベルも15以上になってくるとなかなか現実離れした呪文が飛び交い，マンガの孔雀王みたいになってくる。M&Mのイメージビデオなんか作ったら結構笑えてよろしいんじゃないかな。「オンアボキャベイロシャノ……」などと唱えると，雷や踊る剣が現れて相手を攻撃する図などは，なかなかにアニメチックであろう。

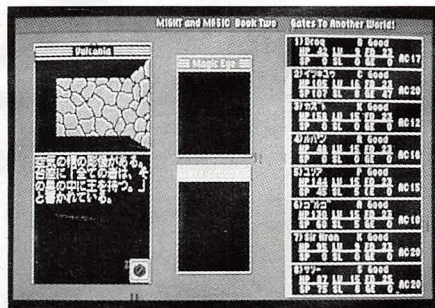
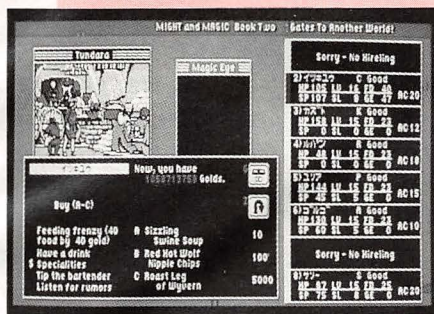
さて，戸外で笑いが止まらない呪文が僧侶の7-3,「Moon Ray(月の光)」だよね。味方に10から100のHPが加わり，敵はダメージを受けるってぇんだからその効果は絶大。町や室内で使ってしまって「戸外じゃないと使えないよ」と怒られ，あせるときもあるほど，私はよくお世話になっている。室内でも使えて凄いのは，魔法使いの8-2「MegaVolts」だ。これのダメージはかなり大きいので，そこらの猛獣や人間の敵なんぞはふっ飛んでしまう。

ただ，こいつの難点はスペルポイント(呪文用のパラメータ)の消費が大きく，2回くらいしか使えないことだ。あと，ジェムを切らして強い敵に会ってしまうと全滅してしまったりするので要注意。ヴォルニカの地下の壁のなかにあるジェムを何度も拾ってたくわえよう。

屋外で150匹とか200匹とかの大量の敵に出会ったときは，魔法使いの8-3「Meteor Shower」が有効だ。うまくすると全部の敵をやってくれる場合もあるが，その数分の間にスペルポイントを一気に使ってしまう。

このくらいのレベルが使えるようなら，戦士が僧侶の呪文の，射手が魔法使いの呪文のレベル5くらいまで使えるようになっているので，「Power Cure」や「Identify Monster」などの低レベルの呪文は戦士や射手に任せておいて，僧侶や魔法使いは凄い呪文を使いまくろう。

もうひとつ笑いが止まらないのが「Duplication」だ。気に入ったアイテムを複製できてしまうのだ。ただ，これには1回100ジェムが必要だが，さっきのところで拾いまくればあっという間に最強装備である。

戦いで重要なのは，盗賊や忍者の忍び寄りで，接近戦の1回目はこれにより強力なダメージを与えることができる。敵によっては男性または女性のどちらかの攻撃しか通用しないものもいるし，「Magic Weapon」

このようなヒントがカギを握っている

これぞ究極のメニューRoast Leg of Wyvern

しか効かないものもいる。その情報は一度会ったらメモすべし。魔法使いの「Identify」で敵の特徴をつかむと同時に，どんな魔法を使うか，金やジェムを奪わないか，などの情報もメモしておかないと，敵のグループのどれを先に倒せばいいのかわからなくなって，負けなくてもいいバトルで大損害を喫することになりかねない。くれぐれもメモはとろう。

究極の経験値の上げ方

経験値を上げて高いレベルの呪文を使えば，戦いもクエストもかなり楽になるはずであるが，実際に経験値を上げるのは非常に時間がかかる。だが，ここにも比較的楽に，しかもテレビを見ながらでも経験値が増える方法があったのだ。こうすれば，うろうろ歩きまわって敵と出会い，やっつけてわずかな経験値を稼いだりしなくてもすむのである。

それには2つのことを知らなくてはならない。まずは食料を売っている店のことである。5つの町の各食料屋では特別料理を売っているが，それは各店に3種類ずつある。これらはぜひ試食してみるほうがいいが，病気になったりスキルが下がったりすることもあるので，食べすぎには注意が必要である。さて，冒頭で述べたラッキードッグサロンの「Roast Leg of Wyvern」というやつが問題のメニューである。こいつは5000ゴールドと高いが，経験値が1500ももらえるのだ。これをジャカスカ食べて，あとはX1をテレビに切り換えてキーを押し続けるだけである。

えっ，そんな金どこにあるんだって？そこが知っていなければいけない2点目である。金のたくさん手に入るところはいろいろあるが，私のお勧めするのはなんといってもD-1エリアだ。え？ご存じない？フッフッフッ，甘い甘い。ほら彫像にも記されていたでしょう。あの「Bozork」がこのへんにナワバリを張っていることが。こいつの部落を襲って（14，7）の地点で見事勝つことができれば，な，なんと！

5億ゴールド！

も貰えてしまうのだ。さあ，一生遊んでくらせるぞぉ。5000ゴールドの10万倍であるから経験値にするとなんと1億5000万だ（そんなにキーを押し続けられないけどね）。2～3分も押し続ければ簡単に1レベルアップである。なあんだ簡単じゃない。私はこれでレベル9から15～16まで一気に上がってしまった。世の中，結局は金だね。

アイテムの使い方のポイント

M&Mの大きな要素として忘れてはならないのがアイテムである。これによってパーティは強くもなり弱くもなるのだが，その設定，取捨選択は意外に難しい。まず，新しい武器を手に入れて，袋がいっぱいになったら近くの町の武器屋で必ずその性能を見てメモをとろう。今回の付録として私が出会ったアイテムの一部の表を載せておく。もっと凄いアイテムもかなりあるのだが，あまり載せてしまうと自分で見つける喜びが半減してしまうから，この表ではごく標準的なものにとどめた。

アイテムは大きく分けると武器・防具・アイテムの3つになる。武器は手に持つ武器と飛び道具とに分けられ，なかには両手がふさがるために楯が持てない「2-handed Weapon」もある。この「2-handed」は普通の武器に比べて強力な場合が多い。表のアイテム名に＋1とか＋5とか付くと与えるダメージがその分大きくなるので，見つけたらそちらに取り換える。また誰がどれを装備できるかということと，魔法のないアイテム（ある怪物には効かない）かどうか，などの違いもあり，そのほか寒さや電気に耐性のある武器などバラエティに富んでいる。これらは一覧表にしておいて，ときどき装備を付け換えよう。防具にも，鎧，楯，ヘルメットの3種類がある。敵のダメージを減らすにはできるだけAC（アーマークラス）を上げておきたい。

アイテムは，いろいろな場所で役に立つ「Key」や「Ticket」の類と，魔法のアイテム，そしてそのほかスキルなどに関連するアイテムの3種類に分かれている。KeyやTicketは持っていないと困るが，「Sextant」とか，どうしようもない魔法のアイテムなんかはその場で捨ててしまったほうがいい。スキルに関するものも必要なだけ取って，思いきってあとは捨てないと冒険の途中ですぐ袋がいっぱいになってしまう。町の旅館にアイテム所持用のキャラクタを待たせておいて渡す方法もあるが，これは意外に面倒である。

魔法のアイテムのなかで使えるのは「Max HP Potion」で，これを何回も使ってHPを上げておき，強い敵と対戦するとクエストレベルの強敵でも意外に楽にクリアできるのだ。Max HP Potionは町で売っているので，そこで買っておき魔法で何度も効果を戻して使い続けることができる。

スキル関連では，「Defense Ring」や「Emerald Ring」を装備すれば誰でもアーマークラスが上がるのだ。Defense Ringは店で，Emerald Ringはツンドラの町の外壁のなかで（ツンドラ新聞や酒場の噂がヒントになっている）手に入れることができる。Emerald Ringのほうは，ぜひ複製して全員に持たせておきたいところだ。

表1　私が出会ったアイテム紹介

アイテム	呪文
Magic Herbs	C1-4
Red Key	—
Red Ticket	—
Wakeup Horn	S1-1
Yellow Key	—
Black Key	—
Castle Key	RN 盗賊能力＋5％
Holy Charm	C1-7
Yellow Ticket	—
Ray Gun	S1-3 正確さ＋5％
Thief's Pick	RN 盗賊能力＋15％
Witch Broom	S3-2
Magic Meal	C3-2
Storm Wand	S3-4 電気への耐性＋10％
Hero Medal	C2-2 人格＋4
Magic Charm	S2-7 魔法への耐性＋10％
Dog Whistle	S4-4 幸運＋1
Admit 8 Pass	—
Sextant	S1-6
Steal Cape	RN 盗賊能力＋10％　速さ＋5
Max HP Potion	HP＋512
Lava Grenade	S4-3
Black Ticket	—
Herbal Patch	C2-1
Dove's Blood	C4-3
Super Flare	C3-5
Defense Ring	S4-5 AC＋2
Acy Gauntlet	KPACRNB 正確さ＋8
Silent Horn	C2-6 毒への耐性＋10％
Emerald Ring	AC＋15

いよいよクライマックス

とにかく凄いのは，なんといってもアトランティウムの8体の彫像が出すクエストであろう。これらはどれもわかりやすい。しかもレベルが高くて，工夫する頭があれば比較的楽なクエストである。8体の彫像はそれぞれ，バードリックヒルの野蛮人，霧の沼地獄の忍者，ファルコンの森の射手，ジョースター街道の騎士，マウントファーヴェーの盗賊，禁断の森の戦士，ロストソウルの森の僧侶，アンシェント島の魔法使い，という8種類の職業に対するクエストを意味している。

それぞれの地名は例の付録マップにしっかり書かれており，それらしい敵の絵もちらほら描かれているではないか。そうかジョースター街道ってのはこのエリアのあのへんか，とたどっていけば，必ず目的の敵に会える。そこで魔法のアイテムやMax HP Potionを使って見事やっつければ，かなりよいごほうびが貰えてしまう。こいつぁ，おいしいぞ。

もちろんこれらのクエストに出会ったり，クエストの目的地に行くのは，結局端から端までしらみ潰しに歩いて地図を描いていくしかないのだ。

さてゲームのクライマックスを迎えるのがヴォルカニアの4体の像(うーむ，また像か）のクエストである。これは水，火，大地，空気の4つの精の像だ。そうか，このマップの4隅にある「Elemental Plane」とかいうのに関係していそうだぞ。しかも僧侶の呪文に，このプレーンに旅するのがあったっけ。ふうむ。そして城のクエストを解いたときのあのヒントは？　そう，この僧侶の4つの呪文はそれぞれのプレーンの近くに隠されているのだ。そしてさらに謎は続いていく。ここからがM&Mのクライマックスなのである。

そのほか各地至るところにさまざまなクエストや，変わった木や泉（スキルが増したり減ったりする）などが隠されている。B-2エリアのサーカス，C-1エリアの騎士マーク，ヴォルカニアの地下などに存在する多数のハイアリング，D-1エリアの礼拝堂，B-2エリアにソリで降りてくる宇宙人，同じくB-2の巨大なマンゴー，B-1(9，9)にいるペガサス（1年の最後の日に開く），A-3エリアの毒の泉，ラクサス城の王女から貰うもの，また格闘の部屋で勝ったときの黒い王冠。

などなど，などなど，まだまだいっぱいあるよ。

最後は根性だ

とにかくこのゲーム，マップもストーリーもハンパではない（スピードの遅さもハンパじゃない!?)。気合を何度入れ直しても，長く長く続く戦いなのだ。それは一歩間違えば人格を破壊してしまうものなのかもしれないし，思いきり挫折感を味わわせてくれるものかもしれない。

しかしRPGってぇのは，概してそういうものなのである。そういう苦労になりふりかまわずぶつかっていくことで，自然にクローンの世界の住人になれてしまうのだ。寝ては起き，起きては寝，「Sleep」や「Disease」のスペルと本当に戦ってこそ，クエストを解いた喜び，レベルが上がった喜び，コンプリートの喜びを真に味わい，ゲームの作者との対話ができるのである。

そうだ，人間は努力をするために生まれてきているのである。努力なしに人生の面白味は理解できない。RPGもしかり，なのだ。諸君の努力に私は期待する。

おおー，やり慣れない話をブツとどっと疲れが出たぁ！　これにて「RPGのプロ養成講座」はおしまい。

表2　防具のアーマークラス一覧

防具	使える人	AC（アーマークラス）
Small Shield	KPCRB	1
Large Shield	KPCRB	2
Padded Armor	—	2
Bronze Helm	KPCB	2　毒への耐性＋15%
Iron Helm	KPCB	2　眠けへの耐性＋15%
Helm	KPCB	2
Cold Shield	KPCRB	3　寒さへの耐性＋15%
Scale Armor	KPACRNB	3
Ring Mail	KPACRN	5
Chain Mail	KPACR	6
Sprint Mail	KPC	7
BSprint Mail	KPC	7　毒への耐性＋15%
SSprint Mail	KPC	7　エネルギーへの耐性＋15%
Plate Mail	KP	8

＊使える人は，すべてキャラクタの頭文字で表しています（表3も同じ）

表3　武器とダメージポイント

武器	使える人	ダメージポイント	武器	使える人	ダメージポイント
Small Club	—	2	Cross Bow	KPARN	8
Dagger	KPASRNB	4	Nunchakas	KN	9
Slumber Club	—	4　眠気への耐性＋15%，呪文S1-7	Scimitar	KPAR	9
Blow-Pipe	KPASRNB	4	Glaive	KPAB	10
Cinder Pipe	KPASRNB	4　呪文S4-3　火への耐性＋10%	Blazing Axe	KPARB	10　火への耐性＋15%
Small Knife	KPASRNB	4	Accurate Smd	KPAR	10　正確さ＋10%
Large Club	—	4	Katana	KN	10
Sling	KPARNB	5	Long Bow	KPAN	10
Cudgle	KPACRN	5	War Hammer	KPACB	10
Short Bow	KPAN	6	Cold Blade	KPAR	10　寒さへの耐性＋15%，呪文S4-1
Maul	KPACRB	6	Trident	KPAB	11
Bull Whip	KCSRNB	6	Great Bow	KPA	12
Long Dagger	KPASRNB	6	Naginata	KN	12
Spear	KPARNB	7	Pike	KPAB	12
Mauler Mace	Neutral-KPACRB	7　強さ＋5	Bardiche	KPAB	13
Staff	KPACSNB	8	Halberd	KPAB	14
Wakizashi	KN	8	Great Hammer	KPACB	14
Long Sword	KPAR	8	Great Axe	KPAB	15
Pirate Bow	KPARN	8　盗賊能力＋10%，正確さ＋15%	Harsh Hammer	KPACB	15　強さ＋3
Shock Flail	Good-KPACR	8　電気への耐性＋15%	Flamberge	KPA	16
Burning Cross Bow	KPARN	8　火への耐性＋15%，呪文C3-5	Ice Sickle	KPAB	16　寒さへの耐性＋15%，呪文S4-1

マシン語カクテル in Z80's Bar

第2回——悲恋の条件分岐——

シナリオ：金子俊一　西川善司
特別監修：浦川博之
イラスト：山田純二

Z80's Barの今宵のお客様は，かの光源氏とはなんの縁もない源光君で，話題はZ80の条件分岐について。マシン語にものをいわせてようこちゃんに迫る光君ですが，果たして，今回は条件が揃わず，恋のフラグレジスタは立たずじまいだったもようです。

♬カランコローン(ドアの開く音)

マスター(以下M)：いらっしゃ……これはこれは源氏の君。ようこそおいでを。

源光(みなもとひかる，以下光)：マスター，酔狂がすぎますよ。

M：なにをおっしゃいます。女泣かせのプログラマって言えば，この店の常連でもあなたをおいていませんからね。あっ，ようこちゃん，いつもの「はちみつレモン」出してあげて。

ようこ(以下Yo)：ハーイ。

光：おや，天女が舞い降りたのかと思いました。もっとも，エプロンを掛けた天女がいればの話だが。

M：(傍白)すぐその気になるんだから。

Yo：いらっしゃいませ。

光：よろしければ，名を明かしてください。

Yo：ご自分でお調べになって。

M：(傍白)こっちは朧月夜の君か……。

光：今夜はお暇ですか？

Yo：ごめんなさい。学校でアセンブラの宿題があって，終わりそうもないんです。

光：それならば私に任せていただこう。こう見えても私はプログラマなのだから。それで，どんな宿題なのですか？

Yo：たしか条件分岐のレポートだったかな。

光：BASICならIF文のようなものですね。

Z80の条件分岐

光：ようこさんはFレジスタ（フラグレジスタ）というのを知っていますか？

Yo：ええ，7月号で長老さんに教えていただきましたわ。たしかCフラグ（キャリフラグ）とZフラグ（ゼロフラグ）だけでしたけど。

光：それなら話が早い。初めのうちはその2つがあれば事足りるし，慣れてしまえば残りのフラグの命令も自然と使えるようになっていると思うよ。

Yo：IF文との違いはあるんですか？

光：IF文では数値などの比較や論理演算の真偽なんかが条件分岐の条件に相当しているわけだけど，アセンブラでの判定条件はすべてFレジスタの各ビット(フラグ)状態によって決まってくるんだ。

Yo：どういうことですか。

光：Fレジスタでは第0ビットはCフラグ，第6ビットはZフラグなどのように各ビットにはそれぞれ意味がある（実際には6本のフラグしかないので，2ビットほど使っていない)。よって第nビットが立っているか降りているか(1か0か)を調べればフラグの状態がわかる。だけど，いちいちこのビットはこーだからなんて調べるのは大変だから，「このフラグが1なら(0なら)こうしなさい」という命令があって，そいつを使うわけだ。

Yo：IF文は「○○が××ならこうしなさい」っていう命令だったもんね。

光：そう，○○はフラグ，××は0か1と決まってしまっているIF文なんだ。

Yo：たとえばCフラグが0なら＊＊＊＊番地に飛びなさいとかですか？

光：そうなんだ。BASIC風に書けば，

```
IF CFLAG＝0  THEN ＊＊＊＊
```

アセンブラなら，

```
JP NC,＊＊＊＊
```

となるんだよ。

Yo：NCってなんですか？

光：ノン・キャリの略なんだ。よーするにCフラグが0ってことね。JPはジャンプと読んで，BASICで言えばGOTO文にあたるんだな。条件を書かなければ，そのままGOTO文になる。だから，条件付きGOTO文と言うほうが正確かもしれないな。

Yo：それじゃあ，条件付きGOSUB文なんてのもあるのかしら。

光：あるある。

Yo：え～，うそみたい。

光：BASICでいうGOSUB文は，アセンブラではCALL(コール)という命令になる。いかにも呼び出すという感じで覚えやすいでしょ。

Yo：うん。

光：Zフラグが1なら＊＊＊＊番地を呼びなさいというのは，BASIC風では，

```
IF ZFLAG＝1  THEN GOSUB
              ＊＊＊＊
```

アセンブラでは，

```
CALL Z,＊＊＊＊
```

になる。

Yo：RETURN命令はないんですか？

光：ちゃんとあります。その名もRET（リターン）だから簡単ね。

Yo：それってプログラムの最後に書くやつじゃないんですか？

光：だからあのRETはプログラムも大きなサブルーチンとみなして，プログラムを呼び出したところに帰るためにあるんです。だから，初心者にはありがちなんだけど，最後のRETを書き忘れてぱーそーさせるなんてのがある。

Yo：あっ，私それやったことある。画面が点滅したりしてとってもきれいだった。

M：ディスクのランプもチカチカっと……。

光：ちがうって。きれいなんて言ってる場合じゃないんですよ。暴走したときはシステムをきっちりリセットしとかないと危ないんです。画面やディスク1枚ですめばラッキーなぐらいで，私なんかCRTCいじくってて，ディスプレイから松脂の香ばしさが漂ってきたことがありましたからね。

Yo：はい，以後気をつけます。

光：では，あなたも1杯お飲みになりませんか。

M：やれやれ……。

命令を組み合わせる

Yo：そういえばELSE文はないんですか？
光：ELSE文自体はない。
Yo：なぁ～んだ。
光：おっと，ナメてもらっては困る。この例を見てごらん。

```
NONZERO
        DEC A
        JP   Z,ZERO
        INC  B
        JP   NONZERO
```

これは，Aから1を引きAが0になったらラベル“ZERO”に飛ぶ。Aが0ではないときはBに1を足してラベル“NONZERO”に飛ぶというものだ。
Yo：どーしてBレジスタに1を足しているんですか？
光：特に意味はない。どんな命令がきてもよいし，なくてもよかったんだけど，なんとなくつけてみたってとこ。ELSE文っぽいでしょ。
Yo：そーね。でもやっぱりBASICのIF文に比べると使いづらいような気がするんだけど。
光：どうして？
Yo：だってフラグに関することしか条件にならないんでしょ。
光：そうなんだけど，CP（コンペア）という命令がある。
Yo：どんな命令ですか？
光：つまり英語のcompare，比較するという意味になる。
Yo：わかった！

```
CP A,B
```

でAとBを比較するのね。
光：残念でした。CPはAレジスタとそのほかのものを比較する命令だから，その場合は，

```
CP B
```

なんだよ。
Yo：なんだか7月号で長老さんが言っていたSUBのときと同じ理由ね。
光：おっ，すごくいいところに目をつけたね。まあ，もう1杯。
Yo：SUBとCPってなにか関係あるんですか？
光：大ありなんですよ。SUBはAから実際に引き算をする命令なんだけど，CPは実際には引き算をしないで，引き算をしたつもりで演算結果だけをFレジスタに入れる命令なんだ。だからA＝55_H，D＝78_Hだとすると，

SUB D でA＝DD_H，CY＝1
CP D でA＝55_H，CY＝1

A＝$3A_H$だとすると，

SUB A でA＝00_H，Z＝1
CP A でA＝$3A_H$，Z＝1

ということになる。
Yo：本当によく似ていますね。
光：それだけじゃあない。命令の種類も同じなんだ。

```
CP (HL),
CP (IX+d),
CP (IY+d),
CP r
CP n
```

がCP命令のすべてなんだけど，SUBもCPをSUBに置き換えただけで結局同じ命令しかない。
Yo：それじゃあSUBとCPを1セットで覚えれば楽ですね。
光：そうかもしれないね。
Yo：(突然)あ～なるほど！
光：どうしたの？
Yo：だから，CPとJPやCALLを組み合わせて使えばちゃんとしたIF文になるのね。
光：すばらしい，そのとおりだよ。たとえばBASICで，

```
IF A=B THEN ****
```

はアセンブラで，

```
CP B
JP Z,****
```

となるんだよ。ほかにはBASICで，

```
IF A>C THEN ****
```

だったら，

```
CP C
JP NC,****
```

になる。最初のうちは大小関係でCフラグの立ち方をよく間違えるけど，よ～く考えればできるし，慣れればほとんど無意識のうちにプログラムを書くこともできるから，しっかり練習しておけば大丈夫。
Yo：う～ん，むずかしいなぁ。
光：たしかにFレジスタって，フラグが立ってるかどうかを気にしてあげないとコロコロ変わるからね。慣れればどうってことないんだけど。
Yo：そういうもんなんですか？
光：そう，昔の人も言ってるよね，「Fレジスタと秋の空」なんて。
Yo：(傍白)コイツあぶな～い。
光：なんか言った？
Yo：いいえ，べつに。
光：ふ～ん，まぁいいか。ところで宿題は終わったからどっか遊び行かない？
Yo：その前に，そういえば私，相対ジャンプっていうのを聞いたことがあるんですけど……。

2種類のジャンプ

光：あっ，すっかりボケてた。相対ジャンプはJR(ジャンプ・リレイティブ)って命令で，その命令がある番地－128～＋127バイト以内ならジャンプができるんだ。
Yo：JPとなにが違うんです？
光：命令がマシン語になったときに1バイト短い。命令の実行速度はJPより遅い。
Yo：あんまり意味がないみたい。
光：肝心なのはここからだ。JPは絶対番地を指定するので，1回アセンブルしてしまってマシン語にすると，ある一定の番地でしか動かなくなってしまう。JRは相対番地(何バイト先に(後ろに)飛ぶか)を指定するので，マシン語でも番地をずらしてもしっかりと動作するんだ。サブルーチンを作るのにはもってこいの命令といえる。
Yo：番地をずらすってどういうことですか。
光：たとえば，5000番地にロードして実行させるように作ったプログラム（サブルーチン）があるとする。ルーチン内のジャンプにJPを使っていればそのプログラムを8000番地にロードして実行させると必ず暴走する。ところが，JRならば暴走しない。そんな風にプログラム(サブルーチン)全部をほかの番地に持っていくことを番地をずらすと言うんだな。
Yo：へぇ～。
光：また，どこの番地にロードしても実行可能なプログラムをリロケータブルなプログラムと言う。Z80である程度の大きさのプログラムをリロケータブルに作るのは，かなりのテクニックがいるんだ。Oh!Xで有名なのはS-OS上の逆アセンブラ“Inside-R”なんかがあるんだけども，初心者では解読不能とまで言われている。
Yo：JRにも条件付きはあるんですか？
光：あるんだけど，JPやCALLに比べると

条件が少ないんだ。ただ，押さえるべきポイントというかNZ, Z, NC, Cはあるから心配しなくていいんだよ。
Yo：(傍白)べつに心配はしてないんだけど。
光：それから，若干マニアックな話になるけど，JPは条件が成立しようがしまいが命令の実行には一定時間がかかるんだけど，JRは条件が成立するとJPより時間がかかり，成立しないとJPよりも速くなる。
Yo：なんだかよくわかんない。
光：そのうちわかるさ。
Yo：そういうもんなんですか？
光：そういうもんです。
Yo：う～ん。奥が深い。
光：さ～て，宿題は終わったかな？　それじゃあ遊びに行こう。
M：あの～宿題はいいんだけど，お店のほうが終わってないんですよね。
光：あっマスター，人の恋路をジャマするようなことをするんですか？
M：人の商売をジャマするようなことをするんですか？
光：しょうがないな。じゃあ閉店まで待ちましょう。
Yo：まだ時間があるみたいだから，なにかご注文はありませんか？
光：そうだね，酔い醒ましにアイスクリームでいいや。

PCとSP

Yo：もしよろしかったらPC(プログラム・カウンタ)やSP(スタック・ポインタ)のお話を聞かせていただけません？
光：ほう，わけありですか？
M：はい，ページの都合です。
光：なるほど。PCは現在CPUがどこのアドレスの命令を実行しているかが常に入っているんだ。だから，PCにアドレスのデータを代入することができれば，実行している番地を飛ばすことができるんだよ。
M：JRやJPはPCにロードする命令とも考えられますね。
光：うん，さすがマスター。
Yo：CALLはどうなんですか。
光：CALLも似たようなものだけど，CALLをした番地を(SP)にセーブしてJPする。そしてRETは(SP)から帰るべき番地をロードして，そこにJPするんだ。(SP)の位置がなんらかの理由でずれていると，元の番地がわからなくなって暴走することもある。極端な話，

```
    JP  ****
```

は，

```
    LD     BC,****
    PUSH BC
    RET
```

などという極悪な方法で書くこともできる。一般的にはメリットがひとつもないのでこんなことはしないけどね。ここで出てきたPUSH(プッシュ)はPOP(ポップ)と対になって疑似16ビットのレジスタを(SP)に待避(PUSH)，(SP)からレジスタに代入(POP)させる命令なんだ。
M：一時的にレジスタを保護する場合に使いますよね。
光：そう。だからPUSHとPOPの数を間違えると上記の例のようにJP命令となってあさってのほうに飛んでいく（世間では暴走すると言う）ことになる。注意するんだよ，ようこさん。
Yo：はい，先生。
光：それから(SP)はメモリ上にあることを忘れてはいけない。ゆえに，

```
    KIKEN
      PUSH AF    ;これひとつでも十分
      PUSH BC
      PUSH DE
      PUSH HL
      JR     KIKEN
```

などとすると暴走する。解説すれば，(SP)がメモリを食い潰してしまうんだ。メモリ上には各16ビットレジスタの中身が山のように書き込まれて，結局プログラム自体も破壊することになるからね(CORE WARSのようだ)。大抵の場合ならプログラムと(SP)はかなり離れた位置にあるのでそれほど気にしなくてよいのだけど，さすがに無限ループはマズイ。もし試す場合はディスクを抜いてからにしてね。永久保証付きで暴走しますから。

Yo：(傍白)家に帰ってやってみよ。(そのまま裏口から帰ってしまう)
光：それからスタックは入れる順番と出す順番が逆になる。これは要注意。たとえば，BCに1234H，DEに6789Hが入っていたとする。

```
    PUSH BC
    PUSH DE
    LD     BC,**** ;レジスタを破壊
    LD     DE,**** ;レジスタを破壊
    POP    BC
    POP    DE
```

とすると，DEには1234Hが，BCには6789Hが入ることになってしまう。よって正しくは，

```
    PUSH BC
    PUSH DE
    POP    DE
    POP    BC
```

で元どおりにBCに1234H，DEに6789Hが入ることになる。というわけでスタックを扱う場合には……，
M：あの，源氏の君，そろそろ閉店なんですが。
光：よ～し，これでお開きにしますか。あれ？　ようこさんは？
M：さっき帰りましたけど。
光：えっ，やだなマスター，どうして教えてくれなかったんですか。
M：やあ，すっかり講義に熱が入ってるようだったんで，お声を掛けちゃ悪いかと思いまして。
光：いや，私としたことが……。またそのうち顔を出すといたしましょう。
♬カランコロ～ン
M：ありがとうございました。

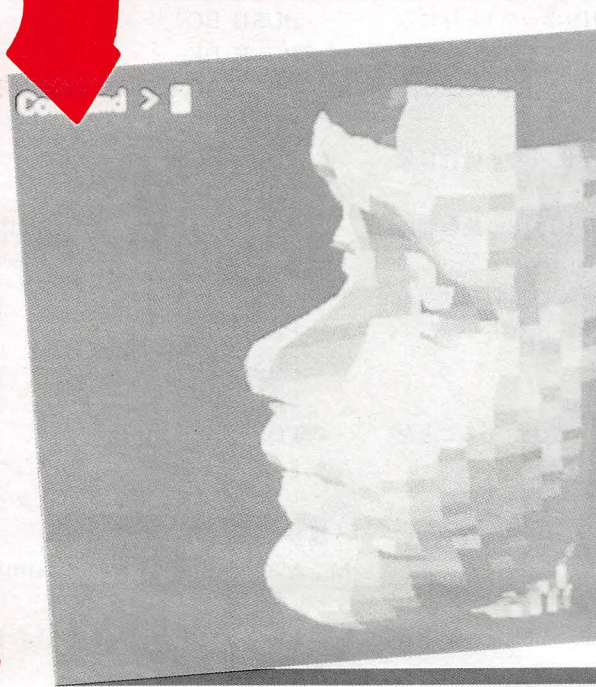
—つづく—

MASTER'S MEMO

○フラグレジスタの各ビットにはそれぞれ名前が付いており，C(キャリ)，Z(ゼロ)，P/V(パリティ・オーバーフロー)，S(サイン)，N(サブトラクト)，H(ハーフ・キャリ)の6つのフラグとX(未使用)2つ。
○JRの判定条件には，NZ，Z，NC，Cがある。JPとCALLにはさらにPO(パリティ・オッド)，PE(パリティ・イーブン)，P(ポジティブ)，M(マイナス)がある。
○CPとSUBの関係を覚える。
○CPと条件付ジャンプの組み合わせでIF文になる。
○判定条件を付けないJR，JP，CALLも存在する(無条件ジャンプ)。
○CALLとRETは対である。
○PUSHとPOPは対である。
○初心者の暴走原因のNo.1はおそらくPUSHとPOPの個数が違うことだと思うので，暴走したらまずこの2つの命令の個数や順番を確認してみよう。
○暴走したあとは必ずリセット（リセットスイッチよりも電源を落とすほうがベター）。偶然にシステムに戻っても，システム自体が破壊されている可能性がある。間違ってもディスクアクセスだけは避けること。

特集2

3Dグラフィックの深淵へ

GRAPHICS

今月は,“3Dグラフィック”の実践に的を絞って，3Dグラフィックをよりリアルに画面上に表示させるためのアルゴリズムを中心にお届けすることにします。まずは，ビデオカメラを使って3Dデータを取り込む「3Dモデリング」。次には「Zバッファアルゴリズム」の後編を。そして最後に，かなり強力なバージョンアップが施されたX68000用レイトレーシングツール，「サイクロンExpressの試用レポート」の3本をご用意しました。今月の記事をご覧になっていただければ，X68000を中心とした3Dグラフィックの環境が，ようやく整い始めたことを実感していただけることと思います。それでは「実際に目に映るものと同様なリアルさ追求するためのアルゴリズムとは?」，その謎を求めて，さらに3Dグラフィックの奥深くへと突き進んでいくことにしましょう。

実践リアル3Dモデリング

Nakano Shuichi
中野 修一
Miyajima Yasushi
宮島　靖

先月は理論編，今月は応用編ということで，ここでは実際に3Dグラフィックを行うときに必ず必要となるモデリングについて考えてみたい。できるだけ簡単にリアルなモデリングを行う簡易3D座標取り込みシステムを，ビデオカメラを使って構築してみよう。

手軽な3D入力を

なんらかの3D処理系を作成したとしてもデータがなければ話になりません。むしろ，3Dデータをいかにして揃えるか，というのが3Dグラフィックのポイントになります。先月軽く前振りしておいたので，3Dグラフィックとモデリングの関係はおおよそ理解していただいたことと思います。今月は実際にモデリングのためのプログラムを取りあげてみましょう。

先月も話したとおり，3Dのモデリングではラフなデザインを元に手作業でデータを作成するという方法が万能なのですが，これでは手間がかかりすぎます。

2Dならば，グラフィックエディタで作ったデータとデジタイズ画像のような2種類の形態がありますね。3Dエディタでデータを作成していくというものはすべての3Dツールに付属している機能ですから，いまさらとやかくいうこともないでしょう。ここでは実際にある物体から直接3Dデータを作成するという方向からモデリングを考えてみることにします。こういった方向からの手法を用いれば，モデリングツールやプリミティブの組み合わせではなかなか実現できないリアルな物体を画面に表示することができます。

といっても，本格的にやると，とかくお金がかかりますので，できるだけ低コストで実現する方法について考えてみたいと思います(あくまでもできるだけ)。

いくつかアプローチ法はあるのですが，ここではビデオカメラを使った3D座標入力の方法とそれを実現するプログラムを作成します。

アルゴリズム

基本的に扱うのは三角測量ですが，実際にどの部分が何cmで……といったデータは必要ありません。各部の比率がわかればよいのです。ここでは大雑把な測量の方法を示しますが，細かな誤差は初めから無視しており，必ずしも正確ではないことをあらかじめご了承ください。

今回のプログラムでは各部の配置に図1のような状態を想定しています。画面に映っている像の中心軸から影までの距離をr，カメラの影への仰角をα，ターンテーブルの回転角をβ(1回の回転角θ×回転回数)とします。すると，物体のY軸(画面と垂直，手前を正とする)の値は，

$$Y=\frac{r}{\sin\alpha}$$

となります。この座標はカメラで取り込んだときの物体の座標であり，Xは常に0です。しかし，本来は角βだけずれた位置にあるはずなので，回転の行列を掛けて変換すると，

$$X=\frac{r\cdot\sin\beta}{\sin\alpha}$$

$$Y=\frac{r\cdot\cos(-\beta)}{\sin\alpha}$$

Z＝画面での高さ

といった感じで求まります。

これを1画面で縦分割した個数×(360/θ)だけやれば，ひとつの物体が完成します。基本は簡単な三角関数と一次変換です。高校生以上の方なら雰囲気はおわかりでしょう。

実践編：まず下準備

まず，最低限用意してほしいものを以下に挙げます。モノさえ揃えばとりあえずなんとかなります。入手方法や品質は問いません。

ターンテーブル
ビデオカメラ
三脚
X68000一式
光源

ターンテーブル（またはろくろ）は画材店やインテリア関係のものを扱っているお店で探すとよいでしょう。今回使ったものは東急ハンズでみつけました(3,200円。東京以外の人ごめんなさい)。

問題はビデオカメラですが，ある程度本格的にやろうという人なら思い切って購入してもよいでしょう。わざわざこのためにカメラを買うのは……という人は電器屋などでレンタルしているものを利用することをおすすめします。運動会シーズンを避ければ簡単に借りられるはずです。もちろん，ちゃんとビデオ信号さえ出ればVHS，Beta Max，8ミリ，Uマチックなど種類は問いません。ただし，ズーム機能はあったほうが（とっても）便利です。

カメラを十分に固定できれば三脚は特に必要ありません。あくまでもあったほうが便利という程度です。

X68000を持っていない人は今回のプログラムは使えません。やっていることは単純ですから，各機種にあわせてプログラムを変更するか，X-BASICコンパチのBASICを作成してください(?)。

光源はできるだけ強力で，はっきりした影のできるものならなんでもかまいません。大型の懐中電灯やOHP（オーバーヘッドプロジェクタ），太陽などがよいでしょう。蛍光灯などは向いていません。

多くの中学，高校にはここで必要な機材のほとんどが揃っているはずですから，学生の方は先生に相談してみるとよいでしょう。

機材が揃ったら，セッティングに入ります。最初はターンテーブルにモーターをつけてX68000から制御するという自動モデリングシステムも一応は考えたのですが，コストが高くなるわりにそれほど精度が期待できません。光線でのスキャンにしても，デジタイズ後，輪郭抽出を行うなど自動化の方向性も考えられますが，角度によってはかなり不確かになったり，高価な光源が必須になるので今回はやめました。よってすべての操作は手動で行われます。

要はターンテーブルの回転角を制御できればよいのですから，ターンテーブルに目盛りをつけてやりましょう。

まず，ターンテーブルの直径を測り円周を出します。次に円周と同じ長さの紙テープを用意してそれをn等分します。いくつでもいいのですが，ここでは36等分にしておきました。これをターンテーブルに巻きつければ準備はほとんど終わりです。図のようにカメラ，ターンテーブル，光源を配置しましょう。

最後にモデリングする素材を揃えます。素材としては，

できるだけ白いもの
ある程度の大きさのあるもの
極端な凹凸がなく中心軸を通せるもの

というのがよいようです。今回使ったのは東急ハンズでみつけた発泡スチロール製の頭部模型，ひらたくいえばマネキン，フランス語読みならマヌカンの頭です。ありきたりのものでは嫌という人は紙粘土かなにかで模型を作るのがよいでしょう。

さあ，これで準備はできました。

プログラムの実行

プログラムはX-BASICで書かれていますので，簡単に理解できることと思います。プログラムの先頭に仰角，回転角，縦分割数などのパラメータを示しておきましたから，自分の目的にあった値に変更して使ってください。たとえば，仰角が45°だったとすると，45°よりも深いくぼみは正確にスキャンできません。逆に仰角が小さすぎると凹凸の状態がうまく読み取れません（精度が悪くなる)。その場の状況でこれらの部分を決定してください。

ここで対象とする3Dデータ構造は7月号で丹氏の作ったものに準拠しますので，まず，あらかじめ先月号の47ページにあるような*.SCNのファイルを作成しておきます。プログラムを実行すると，物体を定義するファイルの名前や色を聞いてくるので，先月号を片手に打ち込んでやりましょう。

それが終わると，いよいよモデリングです。ビデオカメラとスーパーインポーズして固定します。マウスを左右に動かして中

図1

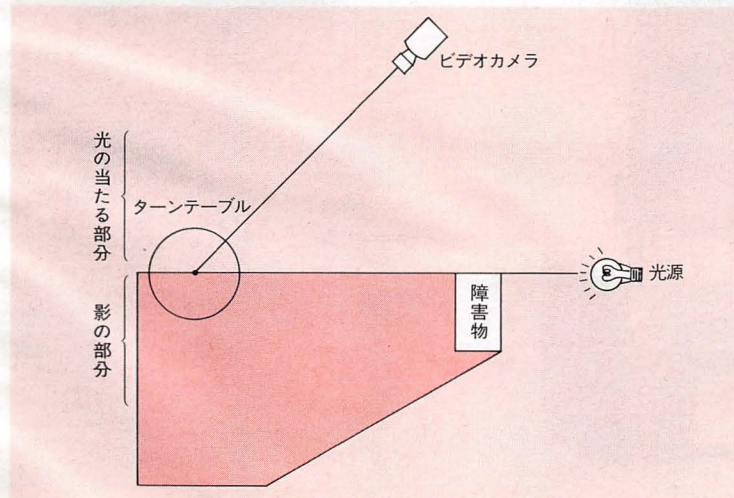



図2

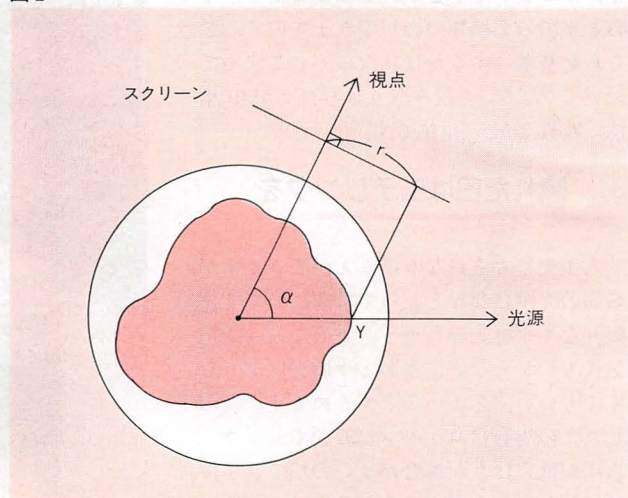
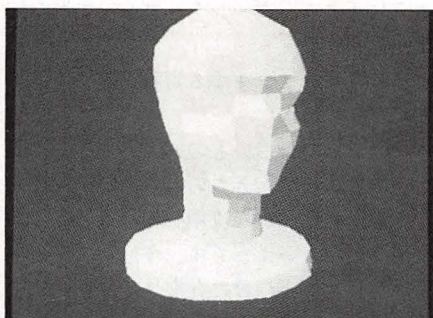



写真1

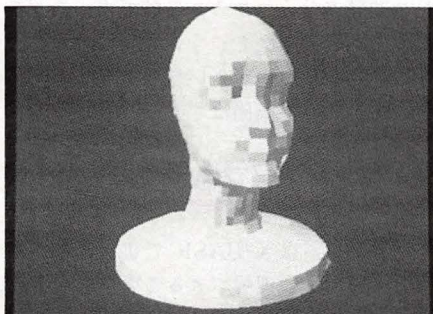



写真2

心軸をあわせて，左クリックします。そして，画面に走っている横線と，物体にできた影との交点を順にクリックしていきます。1画面が終わったら，スペースバーを押してください。ターンテーブルを回転させて次の画面をスーパーインポーズして同じことを繰り返します。1周し終えると，自動的にファイルをクローズして終了します。

光源にいわゆるZライトを使ってみたのが写真1。使用したカメラはシャープXC-85A（たぶんビクターのOEM)。2，3年前の型ですから最近のものよりは性能が落ちるはずです。影が薄く読み取りは困難でしたが，それでもこの程度までなら余裕で大丈夫。読み取りは20°単位で行っています。

写真撮影用の500W電球を使ったのが写真2，3。OHPを使うとかなり理想的な影を得ることができます。その場合でも，本影と半影の2種類の影ができますので，どちらを基準にするかは状況に応じて決めておいてください。読み取りは写真2が10°単位，写真3が5°単位です。

晴れた日はモデリングを

人工光でもそれなりのことができますが，最高の光源はなんといっても太陽光です。限りなく平行光線に近く，誰でも使えて電気代もかからない，本影と半影を区別しなくてもいい，とよいことずくめです。ただし，この場合はロケハンが必要でしょう。太陽が低いほうが都合がよいのでできるだけ早朝か夕方を選んでください。

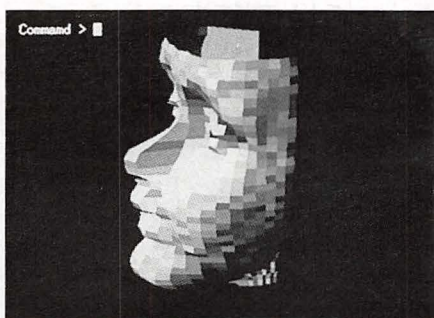



写真3

この場合も機器配置などの要領は同じです。準備ができたらターンテーブルをゆっくりと回しながら撮影していきます。このときは忘れずに下のターンテーブル（の目盛り）まで写しておいてください。

撮影が済んだら家庭用ビデオデッキで再生しながらデータを入力します。つまり影の線がターンテーブルの目盛りと重なった瞬間に画面をスチルし，この状態でポイントをサンプリングしていくわけです。この場合，ビデオデッキにジョグダイヤルがあればいうことなし，コマ送り機能はほしいけど，最低限スチルさえできればなんとかなります。ということは，ごく標準的なビデオデッキでも大丈夫ということです。テープの傷みが気になる場合はカラーイメージユニットでデジタイズしましょう。

とにかく，できるだけ省力かつ安上がりに3D入力をしてみようというのが今回のテーマですから，太陽光線は無駄にできません。晴れた日にはカメラを借りてデータ入力をしましょう。

技術的課題

今回のプログラムではどうしても正確に入力できない物体もあります。複雑なものはいくつかの部分に分割して入力すべきでしょうし，そのためにはデータの移動，回転がサポートされていたほうが望ましいですね。

根本的なところから改善作を探っていくと，ある程度細かいものを入力する場合，データ量が必要以上に大きくなってしまうことが挙げられるでしょう。このプログラムでは全体に一律の処理を行っていますから，局所的に変化量の大きいところではどうしても追従しきれない部分が出てきます。そこを正確に入力するには全体のデータも増えざるをえないのです。

これはポリゴンの頂点間のつながりぐあいを簡単に処理しているための弊害です。限定した用途ならこれでも，そう困ることはないでしょう。しかし，局所的に制御点のサンプリング密度を変えることができるようにすれば，データ量の最適化が図れます。そのようなアプローチとしてNICOGRAPH'87での「体形モデルと双3次スプライン曲面フィッティング」などの発表が参考になるでしょう。より本格的にやろうとする人は参考にしてください。

より少ない制御点で精密なモデリングを行うにはこういったスプライン曲面などでの補間も必要になってきますが，現状のプログラムを生かすならばスムースシェイデ

写真4　今回はこのようなセッティングでデータを取り込んだ

ィングで我慢しておきましょう。

なお，このオブジェクトデータ構造は簡単にCGAフォーマットにコンバートできます。すなわち，

```
NAME attribute ATR
        shading  constant
            :
            :
end
```

というのをED.Xで，

```
prim poly (
    :
    :
)
```

のように変換します。単純な操作ですからエディタのキーボードマクロを活用しましょう。

当然，逆もまた可能ですので，CGAのモデラーで作成したものと合成したりといったデータの相互利用が簡単にできます。今回のプログラムも特にZバッファ用ということではなく，一般的な３Ｄデータ入力の手法として使用できます（多少の改造で大丈夫）。

夢に向けて

今回は現実にある物体からのデータ入力について話を進めてみましたが，コンピュータ３Ｄグラフィックにとってこれはひとつの手法にすぎません。デザイナーが創造力を駆使してモデリングするところに３Ｄの面白さがあるのも事実ですから，あまりこのようなものに頼ると本当の３Ｄの面白さが味わえなくなってしまうかもしれませんね。

将来的には３次元デジタイズ用の安価なハードウェアが商品化される可能性もあります（現状でも商品がなくはないが）。そうなれば，より写実的なモデリングが簡単に行えるようになるでしょう。そのような場合でも，これらはあくまでも補助的に使うのが本筋ではないかと思います。３Ｄグラフィックが描き出すのはある意味で「夢」の世界なのですから。

リスト１

```
  10 /*  カメラ入力簡易モデリングツール
  20 /*                        Y.Miyajima
  30 int jiku=255,kai,fp,poly
  40 int seta=45          /* カメラの影に対する角度
  50 int kaku=10          /* 1回につき回転する角度
  60 int tate=15          /* 縦分割の数-1
  70 float rad=3.1415926535898#/180
  80 dim int px(15),py(15),pz(15),px1(15),py1(15),pz1(15),px2(1
5),py2(15),pz2(15)
  90 str fname,crlf,tb
 100 init()
 110     j_adj()
 120 for kai=0 to (360/kaku)-1
 130     camera()
 140     click()
 150     if kai<>0 then f_wrt()
 160     if kai=0 then {
 170      for i=0 to tate
 180          px1(i)=px(i)
 190          py1(i)=py(i)
 200          pz1(i)=pz(i)
 210      next
 220      }
 230     cpy_dim()
 240 next
 250 f_wrt()
 260 fwrites("END"+crlf,fp)
 270 fclose(fp)
 280 end
 290 /* Initialize
 300 func init()
 310 str p1,p2,p3,p4
 320     crlf=chr$(13)+chr$(10):tb=chr$(9)
 330     screen 1,2,0,1:crt(3)
 340     mouse(4):mouse(2)
 350     apage(1):vpage(3)
 360 locate 3,3
 370     input "ファイル名を入力して下さい";fname
 380     fp=fopen(fname+".OBJ","c")
 390 locate 3,3
 400       input "物体の色を入力して下さい(R,G,B) ",p1,p2,p3
 410       fwrites("ATTRIBUTE"+crlf,fp)
 420       fwrites(tb+"color"+crlf,fp)
 430       fwrites(tb+tb+"DIFFUSE"+tb+p1+" "+p2+" "+p3+crlf,fp)
 440       fwrites(tb+"END"+crlf,fp)
 450       fwrites("END"+crlf+crlf,fp)
 460       fwrites("POLYGON"+crlf,fp)
 470     for i=1 to tate+1
 480         line(0,(i*(500/(tate+1)) and &HFFFE),511,(i*(500/(
tate+1)) and &HFFFE),150)
 490     next
 500     line(jiku,0,jiku,511,150,&H5555)
 510 cls
 520 endfunc
 530 /* 画像取り込み
 540 func camera()
 550      apage(0)
 560      locate 3,3:print"スペースバーを押すと、次に進みます 。"
 570      while(inkey$(0)<>" ")
 580      endwhile
 590      wipe()
 600      cls
 610 endfunc
 620 /* 中心軸の調整
 630 func j_adj()
 640 int x,y,bl,br
 650      msarea(jiku,255,jiku+1,256)
 660      msarea(0,0,511,511)
 670      mouse(2):apage(1)
 680      while(1)
 690          msstat(x,y,bl,br)
 700          if x<>0 then {
 710              line(jiku,0,jiku,511,0,&H5555)
 720              line(jiku+x,0,jiku+x,511,150,&H5555)
 730              jiku=jiku+x
 740              if jiku<0 then jiku=jiku+511
 750              if jiku>511 then jiku=jiku-511
 760          }
 770              locate 3,3:print using "X=###";jiku
 780          if bl=-1 then break
 790     endwhile
 800 endfunc
 810 /* ポイントのクリック
 820 func click()
 830 int i,w,x,y,bl,br,xx=0,xx2=0
 840 /*
 850     apage(0)
 860     mouse(1)
 870     for i=1 to tate+1
 880         for w=0 to 1000
 890         next
 900         msarea(0,i*(500/(tate+1)),511,i*(500/(tate+1))+1)
 910         while(1)
 920             msstat(x,y,bl,br)
 930             mspos(xx,y)
 940             locate 3,3:print using "X=###";xx;
 950             print using "  X=###";xx
 960             if bl=-1 then break
 970         endwhile
 980         calc_3d(abs(xx-jiku),i-1)
 990         if i<>1 then line(xx2,(i-1)*(500/(tate+1)),xx,i*(5
00/(tate+1)),50)
1000         xx2=xx
1010     next
1020 endfunc
1030 /* 3次元座標への変換
1040 func calc_3d(x;int,i;int)
1050 /*
1060 /*
1070 /*
1080     px(i)=(x/(sin(seta*rad)))*sin(kaku*kai*rad)
1090     py(i)=(x/(sin(seta*rad)))*cos(-kaku*kai*rad)
1100     pz(i)=255-i*(500/(tate+1))
1110 endfunc
1120 /* ファイルへの書き出し
1130 func f_wrt()
1140 str·xs,ys,zs
1150 int i
1160 /*
1170     if kai=(360/kaku) then cpy_dim2()
1180     poly=poly+1
1190     for i=0 to tate-1
1200     fwrites(tb+"P"+str$(poly)+"_"+str$(i)+crlf,fp)
1210     fwrites(tb+tb+"ATTRIBUTE"+tb+"color"+crlf,fp)
1220     fwrites(tb+tb+"SHADING"+tb+"constant"+crlf,fp)
1230         fwrites(tb+tb+str$(px2(i))+" "+str$(py2(i))+" "+st
r$(pz2(i))+crlf,fp)
1240         fwrites(tb+tb+str$(px2(i+1))+" "+str$(py2(i+1))+"
"+str$(pz2(i+1))+crlf,fp)
1250         fwrites(tb+tb+str$(px(i+1))+" "+str$(py(i+1))+" "+
str$(pz(i+1))+crlf,fp)
1260         fwrites(tb+tb+str$(px(i))+" "+str$(py(i))+" "+str$
(pz(i))+crlf,fp)
1270         fwrites(tb+tb+"END"+crlf,fp)
1280     next
1290 endfunc
1300 /* 配列のコピー
1310 func cpy_dim()
1320 int i
1330 /*
1340 for i=0 to tate
1350     px2(i)=px(i)
1360     py2(i)=py(i)
1370     pz2(i)=pz(i)
1380 next
1390 endfunc
1400 /* 配列のコピー2
1410 func cpy_dim2()
1420 int i
1430 /*
1440 for i=0 to tate
1450     px2(i)=px1(i)
1460     py2(i)=py1(i)
1470     pz2(i)=pz1(i)
1480 next
1490 endfunc
```

よりリアルな表現を追求するための

Zバッファアルゴリズム(後編)

Tan Akihiko
丹 明彦

先月お届けした基礎編に続いて，今月はいよいよ陰影処理からスムースシェイディングへと，リアルさを追求するための処理を解説していきます。かなり専門的な分野となってきますが，いったいX68000を使ってどこまで表現できるのか，期待してください。

第1部 物体の見え方と表現方法

光源と法線の関係を探る

先月で一応Zバッファアルゴリズムの基礎を終わったことにしたいのだが，あの解説はどうも堅かったように思う。文章も堅いなら内容も堅い。とにかくフレンドリさを欠いていたように思われ，サラっと読むには少々難しすぎる内容であったと少々反省している。実際のところ，初心者があの壁を乗り越えるには，3次元図形を扱ううえでの基礎知識，常識といったものを身につけておかなくてはならなかったのである。

先月の記事では，3次元グラフィックを扱うための基礎知識，さしずめ3次元の座標系やベクトル・行列の取り扱いといったところがネックだったかな。高校2年の「代数・幾何」をクリアしておけば，それほど難しい知識ではないだろうが，数学の苦手な人たちにはきっと苦しい戦いになっただろうし，そんな苦しいことに貴重な時間を費やすよりも，あっさり戦いを放棄してしまったほうが楽だしね。

でも，先月解説した部分をまたここで懇切丁寧に補足しながら先に進めていたのでは，半年経っても完全にフォローできるかどうかは定かでない。だから，「皆さんもわかる範囲でいいから，がんばってね」と，あっさりとエールを送って努力に期待するのであった。もしそれが無理だと思えるのなら，気持ちを新たに今月分だけをいちから読んでもらっても別に構わない。先月の解説とは一応は独立しているので，なんとか読み進めることはできると思う。

先月は，コンピュータグラフィックの専門用語がバラバラと出てきて戸惑った人も多いことだろう。説明する側からすれば，アカデミックな用語がふんだんに使えれば極楽至極なのだが，読まされるほうにとっては，地獄だったかもしれないね。でも，これらの用語は教養として身につけておくのも悪くないだろうし，わかるように努力することが3Dグラフィック＝Zバッファアルゴリズムへの道なのだから，こちらも参考文献を自分で用意するなりして，各自がんばってほしいと思う。

物理的な現象を取り込む

先月の反省をもとに，今回は難易度が変化する文章形式にしてみようと思う。最初はなるべく簡単な言葉を使って説明し，だんだん難解な言葉や数式を交えていく，いわばレベルアップ式解説形式で話を進めることにする。

今回のテーマは，「物体がどう見えるか，またどうやってディスプレイ上にリアルな表現を展開するか」という問題である。扱うことが視覚的なものだけに，日常の感覚で理解できる。これは説明するほうにとっても説明を聞くほうにとっても都合がいいと思う。

前回は座標空間などという，現実からかけ離れたテーマばかりだったので，どうしても数学的，形式的になりがちだった。その反省から，もっと具体的に行うための名づけて「物理的な現象をプログラムへ取り込むテクニック」である。

> レベル1：日常で見る例を出して，説明したい現象がどんなものなのかを理解してもらう。
> レベル2：その現象を深く理解するために，やや物理学的なアプローチで考察を加える。
> レベル3：ここで初めて数式を登場させる。計算に必要な記号や式の意味を説明し，公式みたいな形できちんと書く。
> レベル4：Cでプログラムできるように，順序立てて手続きを書き並べる。

基本的には，このレベル3以降を完全に理解するには高校生クラスの数学や物理の知識が必要だが，自分の理解できるレベルの部分だけを読んでいけば，拒否反応もそれほど出ないと思う。プログラムを組むには，もちろん全部読まなくてはならないのだが，とりあえず説明の流れというか，あらすじというか，とにかく見通しを先に立てておいて，それからじっくりと読み返すという方法で理解してもいいだろう。ま，とりあえずは少しずつ前進していこうではないか。

でも結局は，100の言葉で説明するよりCのプログラムリストを見たほうが早かったりする。千の言葉より1行のリスト。百聞は1行にしかず。失敬，とにかくリストはすべてを語ってくれる。

レンダリング

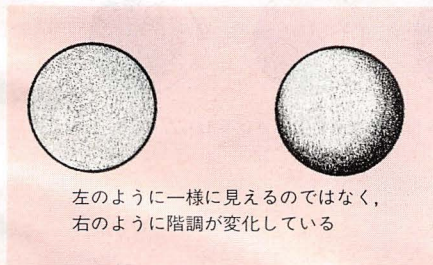
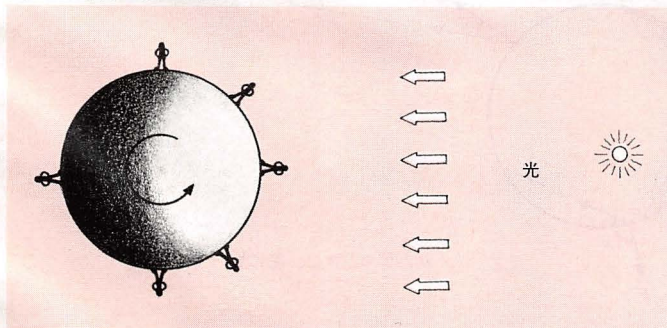
これまでは，いいかげんな色の決め方をしてきたのだが，実はこれではいまいち立体に見えない。そこでリアルな色を付ける工夫が必要になる。先月号のプログラムでは，とりあえず立体に見えるという程度の処理はしてあるが，その理屈もまだ説明していない。

今回のメインは，「レンダリング(rendering，描画・表現といった意味。ここでは特に色彩まわりでのリアルな表現を指す)」と呼ばれる一連の処理である。実は，リアリズムという点ではほかに並ぶものはないといわれるレイトレーシングでも，原理的にはほぼ同じ処理を行っている。

しかし，今回取り上げた手法は，どれも，リアルな表現を目指した近似的な計算方法。物理学的に正確かどうかはさておいて，少ない計算量でいちばん現実感がある表現を目指して，さまざまなテクニックを駆使している。パーソナルコンピュータ用に出ているレイトレーシングのソフトウェアも，多くはこの近似モデルを採用している。現時点ではこれがベストな選択なのだろうと思う。

拡散反射(デフューズ)

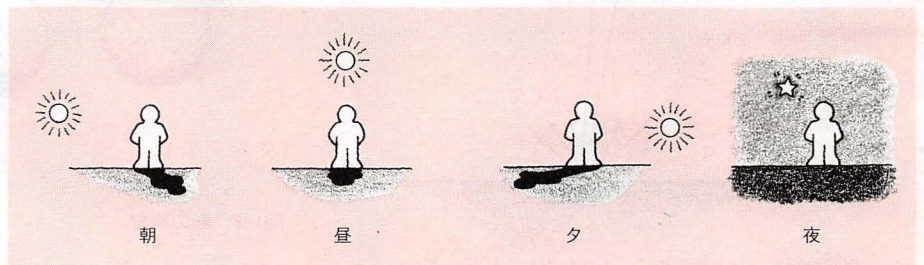
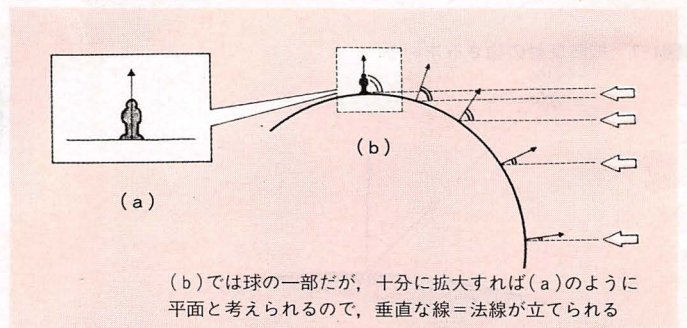
あなたがお持ちのX68000に付いているマウスを手に取って，裏ブタを開けてボールを取り出してほしい。これであなたは僕の説明を理解し，ついでにマウスの掃除もできる。一石二鳥ではないか，などと冗談はともかくとして，取り出したボールをじっと眺めてみよう。ボールはどんな色に見えるだろうか。え，黒？ 「ちっ，こいつは失敗したなあ。僕のはグレイなんだから」というわけで，グレイということで話を続けさせてくれ。果たしてボールはグレイ1色に見えるだろうか？ 決してそうではないだろう。光の当たっている方向は明るいグレイ，そこからだんだんと暗くなっていき，反対側は黒っぽく見えるであろう(図1-1)。

いきなり話をぶっ飛ばす。あなたは，よく晴れた日に運動場の真ん中で1日中立ちっぱなしという罰を与えられた（強引な状況設定ですな)。退屈だけれども，地面を眺めているうちに，あなたはあることに気づくだろう。「朝と昼では地面の明るさが違うな」と。朝のうちはそれほどでもなかったのだが，昼になるにつれて地面はどんどん明るくなり，眩しいほどになってきた。気温も上がってくる。あーあ，なんて暑いんだ（そういえばこれが出る頃には梅雨はもう明けているだろうか)。しかし日射病になって倒れることもなく，無事に夕方を迎えることができた。夕方になると地面はまた暗くなってきて，日が沈む頃には真っ暗になっていた(図1-2)。

夜になってようやく放免されたあなたは，帰る道すがら考える。どうやら光が当たっているというだけでは，地面の色の明るさは決まらないようだ。日が高く昇っているときは明るかったみたいだが，どういうことだろうか。

そこで僕が代わりに答えてあげよう。マウスのボールと地面の明るさは，どうやら光の当たる角度と関係がありそうだ。もう少し正確にいえば，地面にまっすぐ立っていたあなたから見た太陽光の角度は，地面の明るさと密接な関係があるのだ。

地球は巨大な球体である。その地球に太陽からやってくる平行光線が当たっている。地球上には昼のところもあれば夜のところもある。そこに立っている人は，明るいと感じたり暗いと感じたりする(図1-3)。

いま，地球上のある場所に人が立っている。この人を，地面に垂直に立てた線とみなそう(図1-4a)。この，面に垂直に立てた線のことを，幾何学用語では「法線」という。地面の上だけでなく，マウスから取り出したボールの表面にも法線は立てられる(図1-4b)。えっ，曲面の上に垂直な線が立てられるはずがないって？ こんな突っ込みのできるあなたはきっと鋭い人だ。でも僕はもう一枚上手だぞ。なんのために地面と地球などというスケールの大きい話を持ち出してきたのか，勘のいいあなたなら簡単に察することができるね。

さっき法線を立てた地面は完全な平面なのだろうか。落ちている石ころや少々のでこぼこなどは気にしないことにしても，やはり地面は平面ではないのである。ではなにかというと，地球という巨大な球面のごくごく一部なのである。すなわち，曲面も，ごく一部を注目してみれば，ほぼ平面に等しく，そこに垂直な線を立てれば法線になるということがいいたかったわけである。マウスのボールにしても，表面の一部を顕微鏡で覗き込んでみれば，ほとんど平面に違いない(注1)。

ではもう少し専門的な言葉で切り込んでみよう。マウスのボールはグレイに見えるのだが，これは，光源からきた光がボールの表面の細かい凸凹のために方々に散乱するためである（図1-5)。これを拡散反射光(注2)といい，いわゆるツヤ消しの面の見え方のもとになっている。そして，ボール上の1点の明るさに注目して見ていると，どこから見ても変わらないことに気づくだろう（図1-6)。これは拡散があらゆる方向にほぼ均等に起こっていると考えればうまく説明できる。

まとめてみると，曲面上の1点の拡散反射光強度は，「面の法線と光線のなす角度」によってのみ決まり，視点がどこにあっても変わらないということになる。それでは具体的な計算に入ろう。

まずは登場人物の紹介から(図1-7)。面の法線ベクトルをnormal vectorの頭文字を取って「n」とする。光源からやってきて面に当たる光線の方向ベクトルを，lightの

1) この考え方は，数学的には「微分」で決着をつけている。そこまでやると理解できる人なんて全体の1割もいないだろうし（だって理系大学生程度の数学力がいるんだもん)，できたとしても，今回の記事には全然役に立たないので省略させてもらう(ああ助かった)。

図1-1 影の見え方

図1-2 光源と影の変化

図1-3 球体と光の関係

図1-4 法線の考え方

意味で「ℓ」とする。「面から見た光源の方向」を表すベクトルなので，お間違いなく。そして目的の求める量，拡散反射強度「I」は光の強度（intensity）もしくは照明(illumination）の意味。同じように光を当てても，明るく見える物体と暗く見える物体がある。そこで，どのくらい光を拡散反射するか，という指標をユーザーが設定できたほうがいい。これを「k_d」としよう。「d」は拡散反射の意味（diffuseの略)。

先ほどからいっている面法線と光線のなす角をθ(注3)という。略号・記号が多くなってきたので戸惑う方も多いだろうが，ここは気合で乗り切ろう。次はこれら登場人物たちの関係を探っていこう。

拡散反射光強度Iは次の式で求められる。Iは$\cos\theta$に比例する（ランバート［Lambert]の法則)。

$$I=I_L k_d \cos\theta \quad (0^\circ \leqq \theta \leqq 90^\circ)$$

では$\cos\theta$はどうやって求めるかというと，ベクトルの内積を使う。

$$\cos\theta=\frac{(\vec{n}\cdot\vec{\ell})}{|\vec{n}||\vec{\ell}|}$$

これは高校数学の公式だね（高校生の人，思い出したかな？)。さらに，nとℓが単位ベクトル(注4)，長さが1のベクトルだとすると，$|\vec{n}|=|\vec{\ell}|=1$より，

$$\cos\theta=(\vec{n}\cdot\vec{\ell})$$

しかしこれを直接プログラムに組み込むと少しばかりマズい。なぜなら，光線と法線が90°より大きい角度をなす場合は余弦の値が負になる（たとえば$\cos 120^\circ=-0.5$だね）ので，$I=I_L k_d \cos\theta$の値も負になってしまう！　まあ，ちょっと待ってくれ。これはつまり光が当たっていない状態(注5)のことではないか（図1-8)。だからさっきの法則を使ってはいけないのだ。そこで，$\cos\theta$が負になってしまったときは，とりあえず$\cos\theta$をむりやり0にしてしまうと，反射光強度Iも0になり，めでたしめでたしである。

それから，光の当たっていない部分は真っ黒になるはずなのだが，実際はわずかに色が見えることが多い。ボールにしても，影の部分が真っ黒にはならないで，暗いグレイになっていることであろう。これはなぜだろう。

ボールに当たる光は，光源からくる直接的なものだけではない。壁や床などから反射して，間接的にボールに届く光もわずかながらある。これを周辺光(注6)と呼ぶ。周辺光強度は，マトモにやるとなかなかにヘビーな計算を要求する（なにせ環境光のリアルな表現というテーマで論文が1本書けるくらいだから)。そこでわりとうまい近似として，周辺光は物体の場所や面法線，それから周りの物体の配置などにも関係なく「一定」として計算するのが一般的。周辺光も拡散反射光の一種で，その強さは，

$$I=I_a k_d$$

周辺光が起こす拡散反射光の強度は，周辺光の強さI_a（これが一定なわけだね）に比例する。$\cos\theta$が消えただけで，式の形はさっきのとよく似ているね。

ここで例によって例のごとく，手順を書いておこう。

2）向こうの言葉でデフューズ（diffuse）と呼ばれる。散乱反射光ともいう。
3）今月も登場したギリシャ文字のシータ。ついでにいっておくが，今回は角度関係でギリシャ文字があと2つ出てくる（そう嫌な顔をしないで)。α（アルファ）とϕ（ファイ）である。

図1-5　拡散反射

表面の細かい凸凹のために，光が乱反射される。遠くから見ると，あらゆる方向に光が均一に拡散している

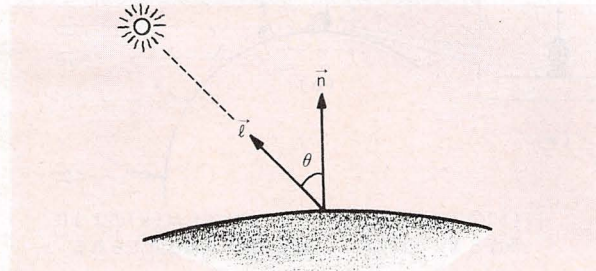



図1-6　拡散反射の強さは視点に影響されない

図1-7　拡散反射の強さを求める

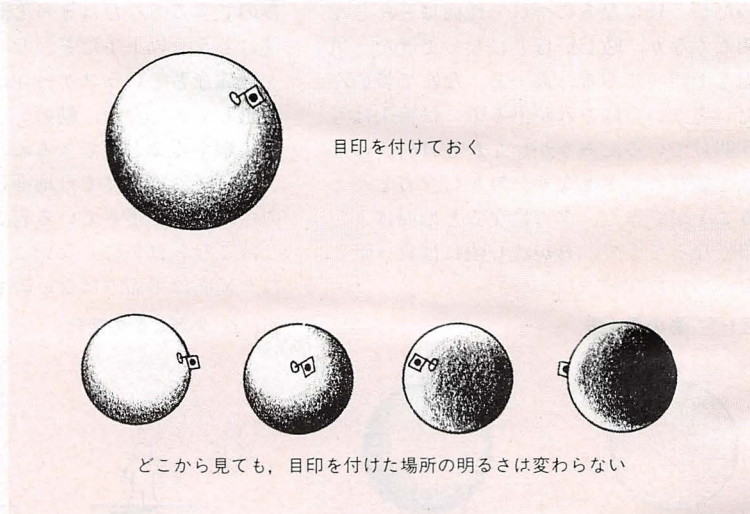
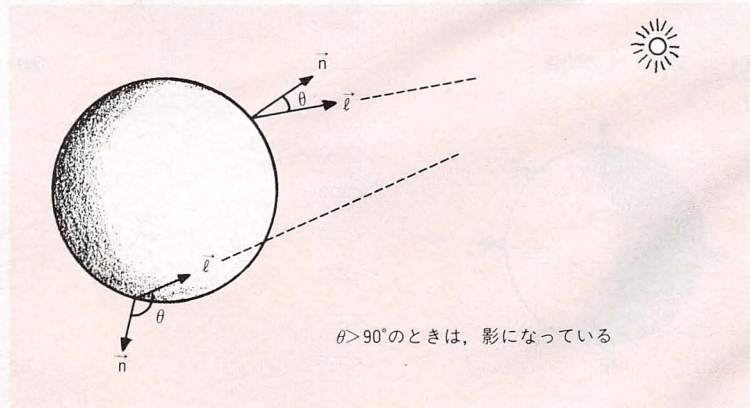



図1-8　影の判定

1） 下準備
グローバルな（シーン全体の）定数：
　光源の強度　I_L
　光源の方向ベクトル　$\vec{\ell}$
　周辺光の強度　I_a
ローカルな（物体ごとの）定数：
　面法線のベクトル　n
　面の拡散反射係数　k_d
を用意する。
　$\vec{\ell}$ と $\vec{n}$ は単位ベクトル化しておく。

$$|\vec{\ell}|=\sqrt{lx^2+ly^2+lz^2}$$

$$\vec{\ell}=\frac{\vec{\ell}}{|\vec{\ell}|}$$

　$\vec{n}$ も同様。
2） 光源からの拡散反射光を計算する。

$$\cos\theta=\vec{n}\cdot\vec{\ell}$$

　もし $\cos\theta<0$ なら，$\cos\theta=0$ と強引に置き換える。

$$I=I_L k_d \cos\theta$$

3） 周辺光の拡散反射光を計算する。

$$I=I_a k_d$$

4） 最終的な拡散反射光強度は，2)，3)で計算した2つの強度の和である。

ただ，このままではモノトーン（白黒）の階調表現しかできない。説明で反射係数に k_d を使ってきたのは，もちろん白黒の照明モデルのほうが説明しやすいからである。しかし，実際のプログラムでは，やはりカラーの表現はしたい。そのためには，k_d をR，G，B別々に指定するとよい。それから光源の色にも当然色を指定できたほうがいいだろう（注7）。こちらは I_L，I_a を同じように細工する。

プログラム中では反射係数 k_d をベクトル化して，c_dif＝(k_{dR}，k_{dG}，k_{dB}）とし，I_L，I_a はそれぞれc_light＝(I_{LR}，I_{LG}，I_{LB})，c_ambient＝(I_{aR}，I_{aG}，I_{aB}）と考えて，上の手順をR，G，B別々に計算させている。ベクトル化しているので，プログラム上はわりとスッキリまとまっている。なお，色のベクトル化については，あとでもう一度解説する。

鏡面反射とハイライト

今日は日曜日，せっかくいい天気なのにドライブにも行かずにせっせと洗車に精を出すお父さん。やりつけないことしてると明日は雨が降るよ，と子供たちにからかわれながらも楽しそうに愛車を磨き上げています。まさか翌日が本当に雨になろうとは夢にも思わないお父さん，ようやくカーワックスもかけ終わったようです。ご覧ください この新車同様の輝き！ 周りを駆け回る子供たちに汚い手で触るんじゃないよと注意しながらも，笑顔がどうしてもこぼれてきてしまいます。

しかし彼はCG屋さんだったので，ピカピカ光る愛車のボディに眺め入っているうちに，周りの風景の映り込みやツヤのほうに気を取られてしまいました（なんて悲しいCG屋さんの性なのでしょう）。

さて，自動車のボディは決して鏡ではない（と，ここでいきなり文体が戻る)。白い車だが，光の当たりぐあいで陰影が付いているところを見ても，拡散反射を起こしていることはほとんど疑いの余地がない。しかしそれでも映り込みはある。太陽の光も反射して目に眩しい。

そこでお父さんハタと気づいた，これが鏡面反射（スペキュラー）なんだと（注8)。鏡でなくても，ワックスを使って十分滑らかな表面に仕上げてあれば，鏡面反射を起こすものである。もちろん鏡のように100%近く光を反射しているわけではない。もしそうだとすると，車は白でなくメッキしたように見えるだろうな（全身メッキした車なんて，考えただけでも，気味悪い)。

さらにあちこちから眺めているうちに，映り込みの様子が見る場所でも変わることに気づいたお父さん，ここで解説は僕にバトンタッチして，さっそくX68000に向かってプログラムを始めた。プログラムしている間，僕がちょこっと数学的に突っ込んでみよう。

僕は小学校で習った覚えがあるのだが，鏡に届く光は，同じ角度で逆の方向に反射する。これと同じことを高校でも習ったのだが，だいぶ言葉は難しくなっていて，「入射角と反射角は等しい（図2-1)」などという表現だったように思う。どうしてそうなるのか知りたい人は，物理学の教科書でも読んでもらうことにして，ここではそれを鏡面反射の表現に利用することを考えよう。

ただ，鏡面反射といっても，今回は光源の映り込み，「ハイライト(hi-light)」しかサポートしない(注9)。自動車のボディに反射していた太陽がそれだ。今回のプログラムでは光源は点光源なので，映り込みも点になってしかるべきなのだが，そこはそれ，むしろ少し広がりを持たせたほうが図2-2を見てもわかるようにリアルになる(注

4） ベクトル $\vec{v}$ の長さは $|\vec{v}|$ という記号を付けて表す。

$$|\vec{v}|=\sqrt{vx^2+vy^2+vz^2}$$

によってベクトルの長さを求めることができる。単位ベクトルとは，

$$|\vec{v}|=1$$

が成立しているベクトルのこと。
　逆に，長さが1でないベクトルを単位ベクトルにするには，まず長さ $|\vec{v}|$ を，

$$|\vec{v}|=\sqrt{vx^2+vy^2+vz^2}$$

で求めておいて，

$$\vec{v}=\frac{\vec{v}}{|\vec{v}|}$$

と，そのベクトル自身の長さで割ってやるとよい。これからも単位ベクトルといえば，この関係を断りなしに使うのでご用心。

5） ここはわからなくていいが，これは「光源に対する自己隠面」と考えてもよい。光の当たっていない部分は，光源の方向からは見えないはずだ（だって光の当たらないところってのが影になるんだから，影の部分が光源から見えたらナンセンスだ)。

6） アンビエント（ambient），環境光ともいう。

7） 僕自身は白色光源以外は使ったことがないのだが……。

8） CGをやってる人間ならこのくらいのことは常識として知っているから，「ハタと気づいた」なんてのはまったくのフィクションです。

9） これは手抜きというよりは一種の技術的な限界である（やっぱり手抜きかな)。いまのところ，鏡面反射を完璧にサポートできるのはレイトレーシング法だけである。最近は鏡面部分のみレイトレーシングで処理する方法などが流行っている。

透視変換アルゴリズムにおける補足と訂正

僕としたことが，先月の説明のなかでいくつかチョンボをしてしまったので，この場を借りて補足と訂正をさせていただく。

まずは，32ページの「左手系」と「右手系」の用語の使い方が逆になってしまっていた。正しくは論理座標系が「右手系」で，デバイス座標系が「左手系」である。申し訳ない。あと，透視変換を行う際の計算式のままだと，奥行き方向に向かっている多角形の辺が，直線でなく曲線となって変換されてしまう。それをプログラムのメインルーチンでは，曲線を直線と考えてエッジの両端をつないでいるので，場合によっては面の前後関係が逆になってしまうことがある。

それを防ぐためには，「正規化透視座標系」という座標系を用いる。詳しい説明はここでは省略させてもらうが，35ページに書かれていた，

$$Z_5=d\cdot z_4/(d-z_4)$$

の計算式を，

$$Z_5=1.0/(d-z_4)$$

に変更していただきたい。こうすれば，エッジがきちんと直線になり，しかもデプスの値が0から1の間に収まる。そしてこの変更に伴い，先月掲載したリストの17，18，591～594行も変更したほうがいいので，今月発表分のものにはその訂正を加えてある。先月のものと見比べて訂正しておいてほしい。

また，前回のプログラムでは整数で処理させるために，Z_SCALE(＝32768）を掛けて，Zバッファの処理を固定小数点演算となるようにしている。これは一応15ビットとなるように考えてそうしたのだが，この値をもっと小さいものに変えると，精度は落ちるが処理速度をアップすることができる。

簡単ではあるが，以上が先月の説明における補足訂正部分である。うっかりミスと説明不足の個所がいくつかあったことをお詫びしておきたい。

10)。

拡散反射光の計算でもたくさん登場人物が出たが，連続ドラマの常として，また新しい人物が加わる予定である。彼らはさながら正義の味方のように，僕らが助けを求めたときにタイミングよく現れて，ピンチを救ってくれることだろう。彼らはおいおい紹介していくことにして，まずはレギュラーメンバーの紹介から。

曲面の1点に光線$\vec{\ell}$が入射した場合を考える（しつこいようだが，$\vec{\ell}$は入射する方向とは逆で，光源に向かう方向のベクトルである)。曲面の法線ベクトル$\vec{n}$も毎回登場だ。もちろん2つとも単位ベクトルである。

さて，入射角と反射角が等しいということだから，これだけの材料（$\vec{n}$と$\vec{\ell}$だね）から反射光が計算できるね。入射光ベクトル（＝光源からの光線ベクトル）と反射光ベクトル$\vec{r}$（新登場，reflectionの略）の関係は，図2-3から，

$$\vec{r}=(2\cos\theta)\vec{n}-\vec{\ell}$$

なお入射角（＝反射角）θは，拡散反射光のときの「法線と光線のなす角θ」と同じθで，その余弦$\cos\theta$も，

$$\cos\theta=\frac{(\vec{n}\cdot\vec{\ell})}{|\vec{n}||\vec{\ell}|}$$

さらに$\vec{n}$，$\vec{\ell}$は単位ベクトルなので，

$$\cos\theta=\vec{n}\cdot\vec{\ell}$$

と，同じ式で求められる。

反射光ベクトル$\vec{r}$も単位ベクトルだったほうがあとあとの計算に便利なのだが，図2-3をよく見ればわかるとおり，わざわざ細工しなくても，最初からちゃーんと単位ベクトルになっているので心配ご無用。

さて，この反射光を伸ばしていって，それが視点にぶつかれば，それがつまりハイライトが見えるということだね。これを式で表すとどうなるか。といっても，手持ちの材料だけでは式に表すのは無理。そこで，いま注目している面の座標（v_x，v_y，v_z）と，視点の座標（$view_x$，$view_y$，$view_z$）にご登場願おう（図2-4)。この2つの差は，視点と面をつなぐ直線，つまり視線の方向ベクトルになる。これを視線ベクトル$\vec{s}$（sightの略かな）としよう。

$$\vec{s}=(view_x-v_x,\ view_y-v_y,\ view_z-v_z)$$

このあと，いつものように$\vec{s}$も単位ベクトル化しておく。

つまり，

$$|\vec{s}|=\sqrt{s_x^2+s_y^2+s_z^2}$$

としておいて，

$$\vec{s}=\frac{\vec{s}}{|\vec{s}|}$$

ちょっと式がごちゃごちゃ出てきたが，反射光（のベクトル$\vec{r}$）が視線（のベクトル$\vec{s}$）に重なれば，その面にはハイライトが現れるというところまで話したんだったね。でも，

$$\vec{r}=\vec{s}$$

ってのは，方向ベクトルが少しもズレてはならないという，かなりシビアな条件だ。ハイライトが点にしかならないので，ディスプレイの解像度が粗いと，ハイライトが全然表示されないことにもなりかねない。こんなことでは，反射ベクトルの計算をするだけ時間の無駄というものである。ここで，さっきいった「ハイライトがわずかに広がったほうがリアルだ」という話を思い起こしていただきたい。つまり，

$$\vec{r}\fallingdotseq\vec{s}$$

のときにハイライトが出るのだと考えようというわけだ。

さて，ここで，現在のところ最もコストパフォーマンスのよい（つまり計算時間のかからないわりにはリアルな表現ができる）近似計算法を紹介しよう。

これはZバッファアルゴリズムばかりでなく，多くのレイトレーシングソフトウェアでも採用されていて，ほぼ市民権を得ているといっていいだろう。物理的に正しいものとはいえない，あくまで経験的に得たモデルであるが，なまじっか物理学的に攻めて描画が遅くなってしまうより，よほどおいしい計算法である。

このモデルでは，ハイライト強度は，完全鏡面反射の方向で最も強く，それから外れると弱くなっていく。つまりハイライトが最も強い点を中心にして広がるという形になる。その広がり方は，面の表面によっても変わる(注11)。この分散の度合いを考慮したモデルは，次の式で表される。

$$I=I_L\cdot k_s\cdot hl\cdot\cos^{hp}\alpha$$

新登場の「hl」と「hp」は，ユーザーが勝手に指定してハイライトの様子を設定す

図2-1　入射角と反射角は等しい

図2-2　ハイライトは広がりを持たせるとリアルになる

図2-3　反射光$\vec{r}$の算出

図2-4　ハイライトの強さを求める

10) 詳しくいうと，これは鏡面反射ではなく，光源からの光を若干拡散させながら反射していることになる。鏡のような完全に滑らかな面ではなく，かといって完全にツヤ消しの面でもない。実は，僕たちの身の回りの物の表面は，この程度の滑らかさを持った面がいちばん多い。だから，本文で解説しているようなハイライトの近似モデルは，非常に現実感のある表現を作り出す。

11) ハイライトが滑らかな表面では強度は急激に落ち込み，ハイライトは鋭いものとなる。粗い表面ではゆっくりと落ち込み，ハイライトはぼやける。

るための定数（だから，指定の仕方によっては，現実感あふれるハイライトができるかもしれないし，非現実的なハイライトができてきてしまうかもしれない)。具体的に役目を解説しよう。まずhlはハイライトの強さ。この値を大きくすると，強いハイライトができる。反射率（ここではk_s）の小さい面にも残る(注12)。hpはハイライトの鋭さ（収束度）で，値を大きくするとハイライトが鋭くなり(広がりが小さく)，小さくすると鈍くなり逆に大きく広がる(注13)。

αは反射ベクトルと視線ベクトルのなす角。

$$\cos\alpha=\frac{\vec{r}\cdot\vec{s}}{|\vec{r}||\vec{s}|}$$

またしても$\vec{r}$と$\vec{s}$は単位ベクトルなので，

$$\cos\alpha=\vec{r}\cdot\vec{s}$$

あ，お父さんのプログラムも完成したようだ。いろんなことをいったけど，うまくまとめてある。というわけで，これまでの復習を兼ねて，ハイライトを求める手順を見てみよう。もう一度，図2-4を見てね。

1） 下準備
グローバルな（シーン全体の）定数：

光源の強度	I_L
光源の方向ベクトル	$\vec{\ell}$
視点の位置ベクトル	view

ローカルな（物体ごとの）定数：

面の位置ベクトル	$\vec{v}$
面法線のベクトル	$\vec{n}$
面の鏡面反射率	k_s
ハイライト強度	hl
ハイライト収束率	hp

を用意する。
視線ベクトル$\vec{s}$は，
$\vec{s}=\mathrm{view}-\vec{v}$
で求める。
$\vec{\ell}$と$\vec{n}$，それに$\vec{s}$は単位ベクトル化しておく。
2） 反射光ベクトル$\vec{r}$を計算する。
$\vec{r}=(2\cos\theta)\vec{n}-\vec{\ell}$
ただし，
$\cos\theta=\vec{n}\cdot\vec{\ell}$
もし$\cos\theta<0$なら，$\cos\theta=0$と置き換える。この場合，ハイライトは存在しない（影の部分にハイライトができるはずがない)。
3） 反射光ベクトル$\vec{r}$と視線ベクトル$\vec{s}$のなす角αの余弦を求める。
$\cos\alpha=\vec{r}\cdot\vec{s}$
もし$\cos\alpha<0$なら，$\cos\alpha=0$と置き換える。この場合も，ハイライトは存在しない。
4） 最終的なハイライトを計算する。
$I=I_L\cdot k_s\cdot hl\cdot\cos^{hp}\alpha$

拡散反射のときと同じく，この計算はR, G, Bの3色に拡張して行う。特に反射率k_sの拡張は重要だ。金属などは映り込みに色が付くので，これを拡張したc_spcをうまく操作してやれば，金属の雰囲気が出せるのである(注14)。ただ，金属でないならハイライトに色が付くことはないので，c_spcの成分をR, G, Bで変えるのは，金属だということを意識したときだけにとどめ，それ以外では3つともそろえたほうが無難である。

シェイディング関数

ここまでに述べた拡散反射光と鏡面反射光（ハイライト）を合わせて，CG界では「シェイディング関数」と呼ぶことがある。次に出てくる透明度との関係から，面自身が発している色という意味で「本体色」とも呼んでいる。プログラム中には「本体色」とコメントを方々に入れてあるが，これはシェイディング関数のことだと思っていただきたい。シェイディング(shading)は「陰影付け」といった意味である。

$$I=I_e k_d+I_L k_d\cos\theta$$
$$+I_L\cdot k_s\cdot hl\cdot\cos^{hp}\alpha$$

透明感

ここまで来たら透明体の表現もしたいところだ。ガラスを通して向こうが見えるというだけのことが，一気にリアルさを倍にしてくれるのだ。さらに，レンズや水晶玉の起こす現象，屈折を取り入れて，ガラスの向こうの世界が歪んで見えたりすると，ああCGやっててよかったなとまで思ってしまう。だが，Zバッファは像空間アルゴリズムであるがゆえに，屈折光を扱うのがとても難しい（注15)。

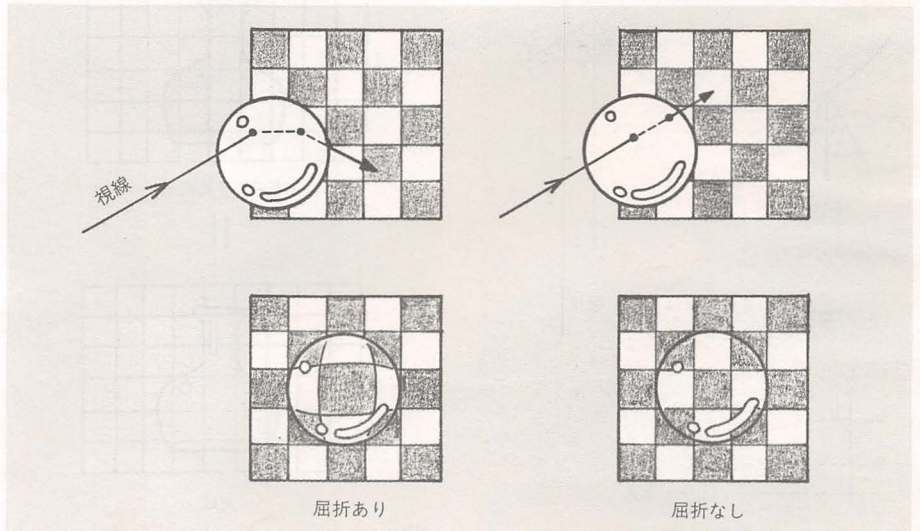
ここでちょっと妥協して，屈折の効果を無視してみてはどうか（図3-1)。こうすると，光線は屈折しないのだから，当然透明体をまっすぐ透過する。これならZバッファアルゴリズムに組み込むときにもホンのちょっとの改造ですむ。

しかし，光をすんなりと通してしまうような物体だったら，それは空気と同じで，目に見えやしない。余談だが，昔読んだ『透明人間』というSF小説の主人公は，服を脱いでしまうと周りからまったく見えなくなった。そこにいるはずなのに目には見えない。これは，彼が自分の体の屈折率を空気と同じにしたためであるが，そういう難しい話はこの際どうでもいい。で，その小説の題名である「透明人間」は，むしろ直訳したままの「不可視人間（原題は確か“invisible man”だったと思う)」のほうが当たっているそうだ。まあこっちの題では意味がわかりにくかったから「透明」としたの

図3-1　屈折は考えない

12）これは光源の明るさに相当する。強い光を当てると，あまり光りそうにないゴムボールや自動車のタイヤでも，ハイライトを作るあれである。

13）タイヤの場合は，ハイライトはできても決して鋭くはならない。このとき，収束度hpの値は1～5くらいの小さな値にするとよい（同時に，強度hlの値も0.2～0.6くらいの小さな値に抑えておいたほうが無難。収束度が低いときは，強度を抑えておかないと，面全体がハイライトのために白くつぶれてしまう)。逆に，磨き上げた自動車のボディは，収束度hpの値を100程度の大きな値にしてハイライトを絞りこんだほうが，メリハリのきいた表現ができる（強度hlの値も10程度の大きさにしよう)。これは実際にモデリングをする人たちへの助言である。参考になっただろうか。

14）鏡面反射光の性質は本当はかなり複雑なものである。反射係数k_sは定数としているが，現実の物質では，入射角や光源の色によっても微妙に変化する。これを加味すると，金属の表現ができる。たとえば銀と鉄とアルミニウムの区別をつけようとするなら，本当はここまで考えなくてはならない。反射率のテーブルを作って対処する方法もあるが，計算時間がぶっ飛んでしまうことだし，ひとまずはお預け。

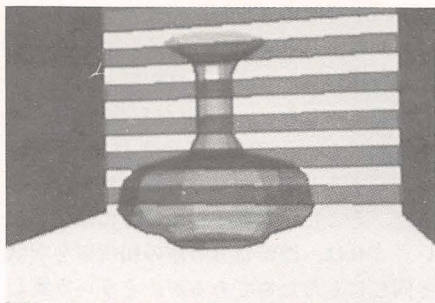
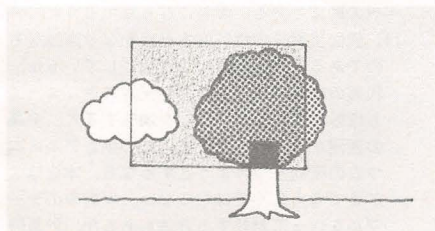
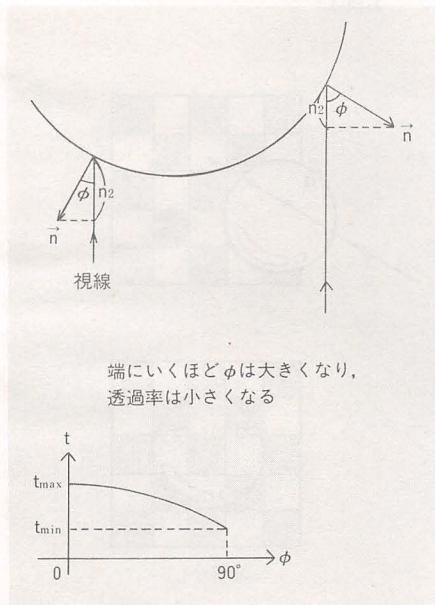



写真1

だろう。

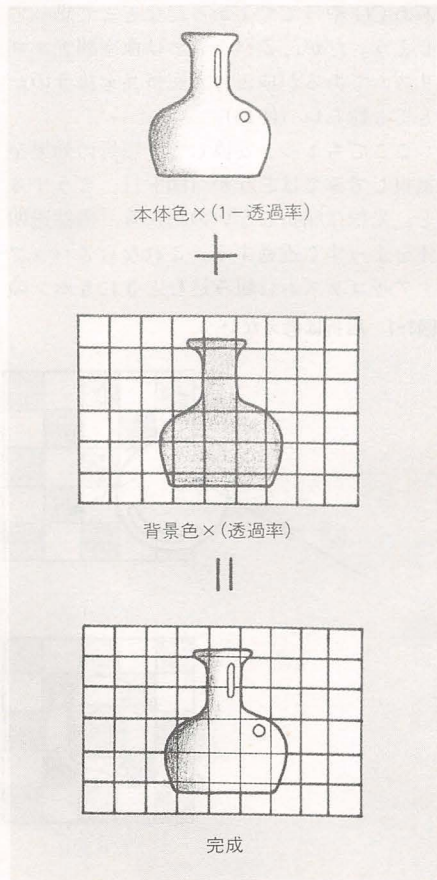
話を戻すが，完全に光を透過してしまう物体は，透明体なんて呼ばずに，不可視体とでも呼んだほうがいい。こんな物体を処理するのは，見えない物体を描こうというのだからなかなかにナンセンスである。やるだけ時間の無駄である。

だから，透過するときに光が少しだけ減衰をするというふうにプログラムを組んでみよう。透過率（光を通す割合。透明度といったほうがわかりやすいだろうか）が100%でなく，もう少し小さい値を取るというわけ。ガラスの向こうの物体が少し暗くなるというだけで，実に単純である。これは物理法則をかなり無視したモデルだが，Zバッファアルゴリズムでも簡単に実現できるし，ソコソコの透明感も表現できる。

これをプログラムするのは実に簡単である。目の前にガラス板をポンと置く。ガラス板の透過率が90%としよう。ガラス板の向こうには，風景が見えているはずだね。だから，その色（明るさ）に0.9を掛けてやる。たったこれだけ（図3-2）。

具体的に説明していこう。この面の透過率をｔ，この面の向こう側（この面に隠される部分）の色をC_{old}とする。新しい色C_{new}は，C_{old}に透過率を掛けて，

$$C_{new}=t\cdot C_{old}$$

で計算できる。

まあこれだけでは，なにか物足りない人もいるだろう。ガラス瓶やワイングラスなどをよーく見ていると，ツヤがあって光っているのに気づくはずだ。そう，ハイライトの存在を忘れてはいけない。しかし，ここでわざわざハイライトの計算をし直すのは，はっきりいってかなり間抜けだ。さっきハイライトの計算の仕方は説明したはずだし，これを利用しない手はない。ここでは一般的に半透明な面を考えてみよう。色は付いているし（拡散反射），ツヤも出ているんだけど（ハイライト），向こうも透けて見える（透明感），そういう欲張りな面である。

登場人物が少し増える。面の「本体色」（説明ずみのシェイディング関数，拡散反射成分とハイライト成分の和のこと）を「C」，透過率を「ｔ」，この面の向こう側の色を「C_{old}」とする。新しい色「C_{new}」は，向こう側の色と本体色をそれぞれ少し弱めて混ぜ合わせた，

$$C_{new}=t\cdot C_{old}+(1-t)\cdot C$$

で表される（図3-3）。

さて，これを直接プログラムに組み込むと，なんとなく味気ないシーンができてしまう（写真１）。花瓶が１枚のガラス板のようにしか見えない。これは透過率ｔを，物体のどこでも一定にしてしまったせいである。この問題を解決するにはどうしたものだろうか。

ガラス板をまっすぐ覗くと，向こうがよく見える。しかし斜めから覗くと，まっすぐのときほど向こうはよく見えない。そこで図3-4を見よう。ガラス板の法線$\vec{n}$と視線$\vec{s}$のなす角度をφ（ファイ）とすると，このことは「φが大きいほど光を通しにくい」と言い換えられる。例によって cosφ と深い関係がありそうなので，それを求めてみよう。cosφは法線$\vec{n}$の視線$\vec{s}$方向の成分$\vec{n}$

15）ここでまとめて，屈折と鏡面反射（映り込み）が，Ｚバッファをはじめとする像空間アルゴリズムでは扱うのが難しい問題であるということを話しておこう。まずは事実関係の整理から。

ａ）像空間アルゴリズムとは，透視変換をかけたあとのデバイス座標系で動作するアルゴリズムのこと。

ｂ）物体をデバイス座標系に持ち込むために透視変換をかけると，物体自身の形や物体の位置関係が微妙に変わる。

ｃ）デバイス座標系のメリットは前回もさんざん指摘しておいたとおり，視線がｚ軸に平行になるので陰面除去がやさしく（＝処理が速く）なるというところである。位置関係が狂うということと引き換えに得た貴重なメリットであるといってもいい。

ｄ）屈折とは文字どおり光が折れ曲がって進む現象である。正確にシミュレートしようと思えば視線を屈折させて，さらにその先にある物体を調べていくことになる。反射にしても同じようにしなくてはならない。

以上のことから，次のことは結論できる。Ｚバッファアルゴリズムで屈折や反射の効果を出そうとするなら，このアルゴリズムの持つ「単純明快さ」，「高速性」，その他諸々のメリットを捨ててかからねばならない。これはまったくもって困った結論である。

というわけで，いまのところ透明体と屈折はレイトレーシングの独壇場である。ただ，最近になってリフレクションマッピング（reflection mapping：鏡面反射して映り込むシーンを物体表面に貼りつける）やリフラクションマッピング（refraction mapping：同じように屈折して見えるシーンを貼りつける）といった手法も出てきて，屈折と反射は必ずしもレイトレーシングの専売特許ではなくなってきている。

図3-2　単純に暗くする

図3-4　縁を強調する

図3-3　本体色を計算する

$\vec{s}$に等しい（ただし$\vec{n}$も$\vec{s}$も単位ベクトルのとき）。

$$n_s = \cos\phi$$
$$= \frac{\vec{n}\cdot\vec{s}}{|\vec{n}||\vec{s}|}$$
$$n_s = \vec{n}\cdot\vec{s}$$

こうしてn_sを求めることができた。n_sが大きいほど透過率は高いことになる。そこで，次のような近似モデルが考え出されている。

$$t = t_{min} + (t_{max} - t_{min})\cdot n_s^{tp}$$

t_{min}，t_{max}，tpはそれぞれ「最小透過率」，「最大透過率」，「減衰係数」。ハイライトのときと同じようにユーザーが指定する。急な角度の面はn_sが小さく，透過率も低い。逆に緩い角度の面では，n_sが大きくなるので高い透過率が出る。これを最小透過率t_{min}から最大透過率t_{max}の間で滑らかに変化させようという主旨である。減衰係数tpの値は，とりあえずは1でかまわない（興味が出たら，いろいろ変えてみてもらってもいいが，それほど大きな変化はない。値を大きくすると，やや縁の部分が強調されるようだ）。

ともかくこの近似式を導入すると，ガラス製の花瓶の縁の部分が暗く，真ん中の部分が明るく見える効果が出せる（写真2）。

それでは透明体をZバッファ上で処理するやり方について検討していくことにしよう。

前のフレームバッファの値C_{old}と，新しく計算した本体色Cとを，透過率tで結合し，新しいフレームバッファの値C_{new}を得る。

$$C_{new} = t\cdot C_{old} + (1-t)\cdot C$$

透過率tは，出てきたばかりの近似式

$$t = t_{min} + (t_{max} - t_{min})\cdot n_s^{tp}$$

で計算しておく。

カラー画像ではR，G，Bの透過率を変えると面白い。色ガラスの効果が出る。透過光，すなわち透明体の向こう側の色C_{old}のみに重み係数k_t（当然R，G，B別々で，プログラム中ではc_trn）を掛ける。

$$C_{new} = t\cdot k_t\cdot C_{old} + (1-t)\cdot C$$

Zバッファの問題点と対策

ここからはZバッファアルゴリズムに特有な問題点である。問題点は大きく分けて2つある。ひとつは，フレームバッファおよびZバッファにポリゴンを書き込む順序，もうひとつはポリゴンのエッジが重なるときの問題。順番に説明していこう。

では，第1の問題。透明体の向こう側の色を，透過光として減衰させたり色を付けたりするのである。もし，先に透明体を書き込んでしまい，そのあとで不透明体を書き込んだ場合はどうなるか？　透明感の効果を狙っても，大失敗に終わるのは目に見えている。こと透明体に関しては，書き込む順序に気を使う必要があるのである。初めに不透明体をすべて書き込でんおいてから透明体を処理しないことには，おかしな画像ができてしまう。

ただ，屈折は考慮しないので，初めに不透明体をまとめて処理しておきさえすれば，透明体を書き込む順番はそれほど関係ない。透明体が何枚も重なると，後ろのシーンはただ暗くなっていくだけなので，ポリゴンの向こう側の色に透過率を掛けていけばすむというのがその理由（図3-5）。さてここで質問。「透明体を書き込むときはZバッファを更新する必要はない。なぜか？」これがわかれば，透明体の考え方はマスターできたといっていいだろう。

第2の問題。エッジが重なるとうまくないことが起きる。特に透明体では，エッジ部だけが重なり，見苦しい表示になることがある（写真3）。そこでちょっと姑息なテクニックを使った。1スキャンライン分の書き込み処理を行う関係zbuff_line( )中では，xでループする際にラストの1点（右端）を処理しないようにしている。結果として，ポリゴンの右端のエッジだけをそっくり描かないことになるわけだ。

こうすると多面体に隙間ができそうなものだが，別に穴が開くこともない。まあこれは，考えてみれば当然のことなのだが。なぜかって？　隣り合ったポリゴンのエッジは共通だね。だからポリゴンを両方とも正直に描くと，エッジの部分には，2回書き込まれるわけ。

写真2

写真3

特に透明体ではZバッファを更新しないので，重複書き込みはよろしくない。だから，重なる2本のエッジを片方しか描かないようにする。こうしても，もう1本のエッジはちゃんと書き込まれるので，支障はまったくない。いやむしろ，片方しか描かないほうがいい結果が得られるのだ（図3-6）。

こういったように，一見手抜きのように見えるやり方のほうが逆にいい結果を招いた場合などはとても愉快である。これだから世の中は面白い。

以上のことを踏まえて，プログラムに少々手を加えよう。拡散反射とハイライトは，本体色を計算しておいてただフレームバッファに書き込むだけでよかったが，透明体は透過率の計算をしておくだけでなく，プログラムの構造そのものにも改造が必要になる。いつものパターンで説明を進めよう。

図3-5　重ねる順序は気にしない

図3-6　エッジの2度描きを省略する

エッジが重なると色が濃くなる

片方の右端は描かない

ポリゴンはつながっている

1) まず不透明体のみ書き込む。半透明体・透明体の処理はその後で行う（プログラム中のポリゴン番号のループの形に注目してみよう。透明体の処理をあと回しにするためにちょいと強引なテクニックを使っているのがわかるだろうか）。
2) 不透明体は前と同じアルゴリズムで処理していく。ピクセルごとにデプスで比較し，Zバッファおよびフレームバッファを更新していく。
3) 続いて（半）透明ポリゴンの処理に入る。先に透過率tを計算しておく。下準備として，
グローバルな（シーン全体の）定数：
視点の位置ベクトル　view
ローカルな（物体ごとの）定数：
面の位置ベクトル　$\vec{v}$
面法線のベクトル　$\vec{n}$
最小透過率　t_{min}
最大透過率　t_{max}
減衰係数　tp
RGBの透過率重み係数　k_t
を用意する。
視線ベクトル$\vec{s}$は，
$\vec{s} = view - \vec{v}$
で求める。
法線$\vec{n}$，それに視線$\vec{s}$は単位ベクトル化して，$\vec{n}$の$\vec{s}$方向の成分n_sを求める。
$n_s = \vec{n} \cdot \vec{s}$
透過率を次の式で計算する。
$t = t_{min} + (t_{max} - t_{min}) \cdot n_s{}^{tp}$
4) 半透明ポリゴンは，ピクセルごとにデプスを比較するところは同じ。もしポリゴンのデプスがZバッファの値より大きいなら，旧フレームバッファ色C_{old}と新しい本体色Cを透過率tを使って混合し，新フレーフバッファ色C_{new}を得る。これをフレームバッファに書き込む。
$C_{new} = t \cdot k_t * C_{old} + (1-t) \cdot C$
（重み係数k_tの右の記号「＊」はカラー化のための特殊な演算。次で説明する）
ただし透明体では，Zバッファは更新しない。

色のベクトル表現について

処理速度のためには望ましいことではないのだが，アルゴリズムが明快になるので，色は（R, G, B）という3次元ベクトル形式を取る。R, G, Bは0.0～1.0の実数値で，表示のときにこれをそれぞれ0～31の整数値に直して16ビットのカラーコードを算出する。

反射係数，透過率などはR, G, B独立になっていて，ベクトル表現になる。当然計算も独立に行うので，色の計算用に特別な演算を用意した。色係数ベクトルの各要素を光源色ベクトルの各要素に独立に掛ける演算「＊」を次の行列式のように定義する(注16)。

プログラムでは関数v_star_v( )，「特殊な演算」というわけのわからんコメントがくっついている部分である。

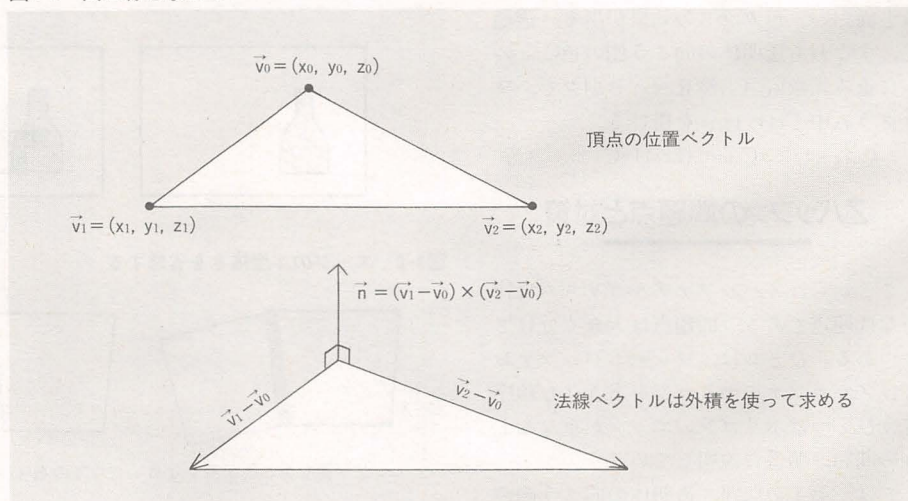
$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} k_R \\ k_G \\ k_B \end{bmatrix} * \begin{bmatrix} l_R \\ l_G \\ l_B \end{bmatrix} \equiv \begin{bmatrix} k_R \times l_R \\ k_G \times l_G \\ k_B \times l_B \end{bmatrix}$$

色をベクトル化したのに伴って，Zバッファの構造体zbuffも拡張する。ピクセルの色（もちろんベクトル表現）とデプスをペアにして，1ピクセル分の属性にしている。スキャンラインと交わるすべてのポリゴンを書き込み終わるまで，処理はこのzbuff内で行う。デプスはもちろん，色も直接画面に書き込まないでzbuffのなかのメンバーに格納する。処理が終わったら，まとめてピクセル色をフレームバッファにコピーする。

処理がストレートでない（直接フレームバッファに書き込むわけではない）ために多少スピードは落ちてしまったが，透明体の表現が可能になったメリットは大きい。

面法線の算出

もうすでに気がついている方がいるかもしれないが，僕はこれまで肝心な部分を意識的に避けつつ説明してきた。拡散反射，ハイライト，透明感のどの説明にも登場したレギュラーメンバー（主人公といってもいいかな）である，面の法線ベクトル$\vec{n}$の身元を全然明らかにしていないのだ。これがドラマなら，出身を隠す主人公などはゴロゴロいるが，あいにくここはシビアなコンピュータのプログラムの世界。正体の知れないものを扱うことなどできようはずもない。

というわけで，ポリゴンの法線がわかったなら，ポリゴンの色はもうあなたにも求めることができる。まだ教えていない面法線の求め方をこれからお教えしよう。

前回のプログラムでは，すでに散乱反射のみの表示まではできあがっていた（理屈は説明してないが)。そこで前回発表のファイルフォーマットを見ると，おやおや，頂点の位置ベクトルはあっても法線ベクトルは見当たらない。ということはつまり，位置ベクトルの情報だけで，法線ベクトルが計算できるということだね。

さて，物事はいちばん単純なところから始めよう。頂点が3つの場合(注17)，三角形からいってみよう(図4-1)。

3頂点の位置ベクトルはそれぞれ，

$\vec{v_0} = (x_0,\ y_0,\ z_0)$
$\vec{v_1} = (x_1,\ y_1,\ z_1)$
$\vec{v_2} = (x_2,\ y_2,\ z_2)$

である。データファイルを見ても，

$x_0\ \ y_0\ \ z_0$
$x_1\ \ y_1\ \ z_1$
$x_2\ \ y_2\ \ z_2$

とだけ書いてあるはずだね。

いささかくどいが，これから求めようというのは三角形$v_0 v_1 v_2$の法線ベクトル，

$\vec{n} = (n_x,\ n_y,\ n_z)$

なのだ。いきなり結論からいっちゃうと，

$\vec{n} = (\vec{v_1} - \vec{v_0}) \times (\vec{v_2} - \vec{v_0})$

なわけだな。おっといきなり計算式攻撃に出てしまった。でも，怒らない怒らない。まあ，この式をいきなり見て理解できた人(要するに理系の大学生)は，ここは読み飛

16) ちなみに数学的にはこういう演算はない。このテの演算は通常「変換」ととらえられるので，行列を使うのが通例であるが，メモリ効率と計算速度のことを考えると，行列はあまりおいしくない。
17) 多角形が成立するためには，最低でも3つの頂点が必要である。だって「二角形」なんて聞いたことないでしょ？

図4-1　面法線を求める

ばしてもいい。理解できなかった人には，これから説明する範囲でなんとか理解してほしい。

つまりこの式は「ベクトルの外積」ですな。いったいどうしてこれで法線が求められるのか，という疑問は当然湧くだろうけれど，僕は今回この理屈をくどくどといいたくはない。とにかく，平面上の3点から法線を求めるには，ベクトルの外積を取ればいいのだと覚えておいてほしい。こいつはまさに天下り論法ですね。あら，怒っちゃった？ まあまあ，抑えて抑えて。

ところで，もしあなたが高校2年生以上なら，この3点から平面の方程式を求めてみてもらいたい。平面の方程式の一般式は

$$ax+by+cz+d=0$$

だから，(x, y, z) に $\vec{v_0}$, $\vec{v_1}$, $\vec{v_2}$を代入すれば，

$$ax_0+by_0+cz_0+d=0$$
$$ax_1+by_1+cz_1+d=0$$
$$ax_2+by_2+cz_2+d=0$$

という3元連立方程式になるね。これを解けば，係数a，b，c，dが出てくるが，平面の法線ベクトルは，

$$\vec{n}=(a, b, c)$$

に等しい（これを知らないとはいわせないよ）。

とにかく，これで三角形の法線は求められる。四角形，五角形などでもまったく手順は同じ。3つの頂点だけを代表として取り出して，この三角形の法線ベクトルを求めればいいわけだ（図4-2）。

ついでに，面の表裏を決めるのにも法線ベクトルを使うので，このやり方も説明しておこうね。

Zバッファアルゴリズムはサーフィスモデルを扱う。多面体のように見えても，正体は独立した多角形の集まり。おまけに，表示したいオブジェクトは閉じた多角形とは限らない（サンプルの花瓶をご覧いただきたい。口の部分が開いているでしょ)。だから，先月号で三沢氏が解説されていた，頂点の連結順序で法線ベクトルの方向=面の表方向を決定し，その表が向こう側を向いている面は表示しない，という「後面除去」の手法は使えない（後面除去が使えるのは，基本的にはソリッドモデルだけ)。

ちょっと脱線したが，今回のプログラムでは，視点から見えるほうの面が表になるように表向き法線の方向を決めることにする（図4-3)。視線の方向ベクトルを，1番目の頂点v_0を代表に立てて，

$$\vec{s}=view-\vec{v_0}$$

で定め，あとは$\vec{s}$と$\vec{n}$の内積，

$$\vec{s}\cdot\vec{n}$$

の符号が負なら，法線はあっち向き。だから，

$$\vec{n}=-\vec{n}$$

とやって，表裏をひっくり返す。

これで，あらゆるポリゴンの法線は，全部こっちを向いていることになる。レンダリングをかけるのは，ポリゴンのこちら側を向いた面なのだから，これでいいのだ。

こうして求めた法線ベクトルは，ポリゴン構造体のなかにしまい込むわけだが，これを，あとの関係もあるので，「真の法線」と名づけておく（ということは，ウソの法線があとで登場するわけだね)。

総まとめ

1) ファイルから頂点を読み込むときに，最初の3点を$\vec{va_0}$, $\vec{va_1}$, $\vec{va_2}$として別に持っておく。「va」などと「a」が余分に付いているのは，プログラム制作者（僕のことだよ）の都合で，ほかのベクトルと区別する必要があったからだ（これは，無計画なプログラミングの報いともいえる)。
2) 平面上の2本のベクトルを作る。2本の異なるベクトルだったらなんでもいいので，次のようにする。
$$\vec{va_1}=\vec{va_1}-\vec{va_0}$$
$$\vec{va_2}=\vec{va_2}-\vec{va_0}$$
3) この2本のベクトルから，真の法線ベクトル$\vec{na}$を，外積を使って作る。
$$\vec{na}=\vec{va_1}\times\vec{va_2}$$
4) 表向き法線の方向を決定する。まず視線の方向ベクトル，
$$\vec{s}=view-\vec{v_0}$$
を求める。
5) $\vec{s}$と$\vec{n}$の内積の符号を見る。もし内積が負の値
$$\vec{s}\cdot\vec{n}<0$$
なら，
$$\vec{n}=-\vec{n}$$
を行って，表方向を決める。

第2部 平面を曲面に見せる

スムースシェイディング

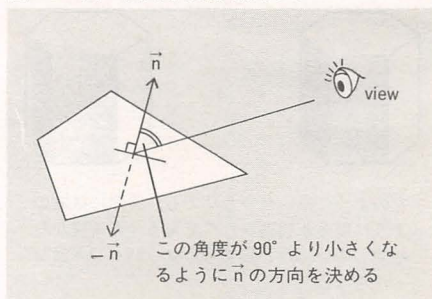
改めて指摘するまでもないと思うが，Zバッファアルゴリズムは，ポリゴン（多角形）向けに開発された手法である。この生い立ちからもわかるとおり，ポリゴンの表示には単純明快で高速な処理を約束してくれる。ポリゴンの数が増えてもその頼もしさは少しも変わらない。いや増えたときこそZバッファは威力を発揮するのである。

多面体にはこれほど強いZバッファ法も，いざ曲面を表現しようとすると，一転して非常に苦しい立場に立たされる。これは痛い。かなり致命的な弱点といってもいい過ぎではない。それではと細かいポリゴンをたくさん用意して曲面の形に並べると，少しは曲面らしく見えるが，それでも不自然さは拭えない。生半可な細分の仕方では，細かい多面体にしか見えないのである。

人間の目というものは妙なところで敏感である。残酷なまでに敏感だ。人間がこの多面体を見ると，ポリゴンの継ぎ目にマッハバンド（注18）が発生するのだ。そのためエッジが強調されてしまう（図5-1)。だから多面体の集まりのような印象が消えることはない。多面体を曲面に見せるのには少々無理があるようだ。Zバッファアルゴリズムは，このウィークポイントのせいで，曲面プリミティブを簡単に扱えるうえに緻密な表現も自由自在にできるレイトレーシングに，その地位を1歩も2歩も譲ってしまったのだ。

18) 多面体表現では，ポリゴンの継ぎ目で明るさが階段状に変化する。その変化がたとえわずかなものであっても，人間の目はそれを見事にとらえる。変化しているところを，実際より極端に変化しているように感じるのである。この現象を発見者の名前をとって，マッハバンド（Mach・band）と呼ぶ。「バンド」とは「帯」のこと。曲面が曲面に見えずに，帯の集まりのように見えるところからきている。マッハバンドは，人間の視覚心理的効果なのだが，ともするとレンダリングの結果を台なしにしてしまう非常に厄介な相手である。

図4-2 三角形でない場合

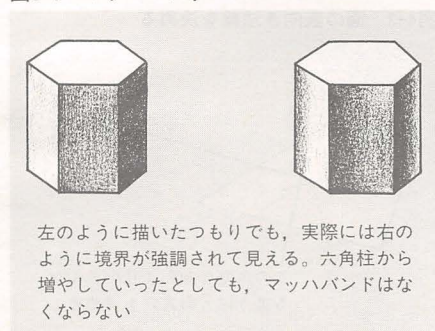
図4-3 面の表向き法線を決める

しかし，捨てる神あれば拾う神ありで，シェイディング補間法（スムースシェイディング）という画期的なアルゴリズムが提案されて成功を収めている。そしてこのZバッファアルゴリズムの場合，嬉しいことにスムースシェイディングの実現はプログラムの少しの拡張で済むのである。ただし説明は多少難解なので，よく読んで理解してほしい。

ポリゴンの色がどこでも同じになる表現，ふつうの多面体表現を「コンスタントシェイディング（constant shading）」という（注19）。コンスタントシェイディング表現で表されたモデルは，エッジの部分で色が不連続に変化しているので，小平面の集まりに見える（マッハバンド）。スムースシェイディングは，この不連続性をなくし，平面を曲面に見せる方法であるといってもいい。

2つのシェイディング補間法

これから2つのシェイディング補間法を紹介するが，両方に共通する基本コンセプトはこうである（図5-2）。たとえば多角形のある頂点では明るい色で，別の頂点では暗い色だったとしよう。仮にその間を明るい色から暗い色に滑らかに変化させたらどうだろう。その多角形はもはや平面には見えないことだろう。何枚もの多角形に渡って色を滑らかに変化させていけば，この多角形の集団は曲面に見えるはずではないか？　そのとおりである。それがスムースシェイディングの考え方の基本なのである。

もう少し難しい表現を使おう。まず，ポリゴンの各頂点について属性(たとえば色)を別々に設定しておく。そして，ポリゴンの内側で頂点属性を滑らかにつなぐ（これを「補間する」という）のだ。

では具体的にはどうやるか。曲面が平面と大きく違うのは，法線ベクトルが場所によって変化するところだ。そこで，ポリゴンの頂点に仮想的な法線を設ける。頂点の間で法線が滑らかにつながってくれれば，平面が曲面のように見えても不思議ではない（図5-3）。

スムースシェイディングとは，ポリゴン上の法線にゆらぎを与えることで，曲面と同じ効果を出し，平面を曲面に見せているのだ。はっきりいってしまえばゴマカシなのだが，そのわりには素晴らしい効果があるので許してあげようではないか。

それではシェイディング補間法の2つを紹介する。それはグロー（Gouraud）シェイディングとフォン（Phong）シェイディングだが，先に発表されたのはグローシェイディングである。とりあえずこの2つの原理的な違いだけを簡単に紹介しておく。具体的な長所や短所，それにプログラムとして実現する手順は追ってあとから解説する。

グローシェイディング

いちばん簡単にいうと，頂点での色を補間する方法である。法線の補間は行っていないので，やや正確さに欠けるものの，けっこう「使える」技法である。

先ほどから述べているように，頂点ごとに仮想法線が設定されているので，その頂点での色を求めることは当然できる。グローシェイディングは，頂点と頂点の間で色を滑らかにつなぎ，曲面らしく見せる方法である。

今回のプログラムでは，ポリゴンの本体色だけでなく，透過率も補間している。

フォンシェイディング

グローシェイディングでは色を補間したが，より正確に曲面を近似するために，フォンシェイディングではまず仮想法線を補間する。各ピクセルを処理する段階で，補間した法線を使って本体色および透明率を計算するのである。ハイライトや透明感の計算には，面の位置ベクトルも必要なので，仮想法線ベクトルと同時に位置ベクトルも補間している。

2つのスムースシェイディング補間法には，お互いに長所と短所がある。

まずグローシェイディングの短所のなかでとりわけ目につくのは，ハイライトが変になることだろう。これは写真の赤い球を見れば一目瞭然である。しかしその理由は少々厄介になる。結局はグローシェイディングが，「色」を補間する方法だからなのだ

19) 今回のプログラムでは，実際のレンダリング処理（関数rendering( )）の出力は本体色と透過率（c と t）である。コンスタントシェイディングではこの2つがポリゴン中どこでも均等に書き込まれるものだと考えてほしい。色だけでなく，透過率も均等ということ。

図5-1　マッハバンド

左のように描いたつもりでも，実際には右のように境界が強調されて見える。六角柱から増やしていったとしても，マッハバンドはなくならない

図5-2　スムースシェイディングの考え方

図5-3　頂点間の法線を滑らかにつなぐ

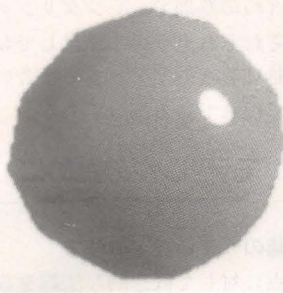



図5-4　グローシェイディング

写真4

が。ハイライトは鋭く，急激に変化する。対して拡散反射はゆっくりと変化する。この両方をひとまとめにして補間するのが間違いのもとなのだ(注20)。

グローシェイディングは，基本的に拡散反射成分を補間するのには向いているが，ハイライトには向かない。この欠点を補う意味で開発されたのが，「法線」を補間するフォンシェイディング。ハイライトを使いたい場合はフォンシェイディングをすすめておく。

ではフォンシェイディングに欠点がないのかというと，やはりある。グローシェイディングに比べて処理が遅い(注21)。ハイライトがなくていいなら，グローシェイディングがお勧め。

困ったことに，どちらにも共通する欠点もある。写真4の物体の輪郭線を見ていただきたい。ここで，スムースシェイディングの化けの皮がはげてしまう。曲面に見せているようでも，所詮は多面体，輪郭は多角形ではないか。この解決方法としては，多角形がなんとなく曲線に見えるところまでポリゴンを細かく取っていくしかない。これではときどきメモリ不足を招くので不経済だ。完璧な曲面がほしいなら，自由曲面やレイトレーシングのメタボールなどがある。が，いずれも重い処理になることは覚悟しよう。

もうひとつの欠点と関連するのだが，凹多角形を使うのは避けてほしい。なぜなのかは写真3を見てもらうとわかる。シェイディングの結果が不自然になってしまうのだ。これはフォンシェイディング，グローシェイディングを問わず起こる。なぜ不連続になるのかは，ちゃんと理由もある(注22)。とにかく，スムースシェイディングと凹多角形は相性が悪い。これだけは覚えておいてほしい。

ついでに，オブジェクトはできるだけ三角形だけで作ったほうが無難だともいっておく。四角以上だと，やはり不連続なシェイディングになる危険性があり，それでは見栄えがよくない。

実現化のためのアルゴリズム

具体的なアルゴリズムを，手順の形でまとめて書いておこう。3つのシェイディング技法を比べると，きっとアルゴリズムの本質が見えてくることだろう。

スムースシェイディングでは，スキャンライン順に処理する関係上，補間のやり方が少しテクニカルである。図5-6にグローシェイディングの例を出したが，フォンシェイディングも似たようなもの。

コンスタントシェイディング

1) ポリゴンの本体色cと透過率tは一定なので，ポリゴン構造体に格納する。これらを求めるには，引数に，

位置ベクトル$\vec{v}$：1番目の頂点$\vec{v}a_0$

法線ベクトル$\vec{n}$：真の法線$\vec{n}a$

を代入して，レンダリング処理ルーチンを呼び出す。

2) メインルーチンは単純で，cとtをそのまま使ってピクセルに書き込む。

グローシェイディング

1) ポリゴンの各頂点での本体色cと透過率tを求める。

その下準備として，i番目の頂点$\vec{v}_i$とそこでの仮想法線$\vec{n}_i$を読み込む。仮想法線$\vec{n}_i$は，ファイルから読み込んだものをそのまま使う(もちろん単位ベクトル化はする)が，視点から見てこちら側を向いていることが好ましいので，真の法線$\vec{n}a$(前に述べたが，視点のほうを向いている)に近い方向を向く(なす角が90°以下)ように修正する。具体的には，$\vec{n}_i$と$\vec{n}_a$の内積の値が負，

$\vec{n}_i \cdot \vec{n}_a < 0$

なら，$\vec{n}_i$の方向をひっくり返す。

$\vec{n}_i = -\vec{n}_i$

真の法線は，頂点を3つ読み込み終わったときにしか計算できない。だから実際には，この法線の方向を修正する処理は，頂点を

20) これを難しくいうと，線形性と非線形性という話になる(ここはわからなくても気にしないでね)。ハイライトは非線形性が顕著だが，グローシェイディングは色を線形補間するものだから，ハイライトの急激な変化についていけないのだ。

21) 仮想法線ベクトルと位置ベクトル，合計2個のベクトルを補間しているうえに，ピクセルごとにいちいちレンダリング処理ルーチンを呼び出して色を求めている。だから，処理が重いのも当然とはいえる。ちなみにグローシェイディングでは，レンダリングは前処理で1回だけだし，ループの中でも，1個の色ベクトル(本体色)と1個の実数(透過率)を補間するだけ。

22) どちらのアルゴリズムでも，スキャンラインに沿って補間するだけで，ポリゴンの形は考慮されていない。そのため，スキャンラインの間でシェイディングが不連続になってしまうことがあるのだ。

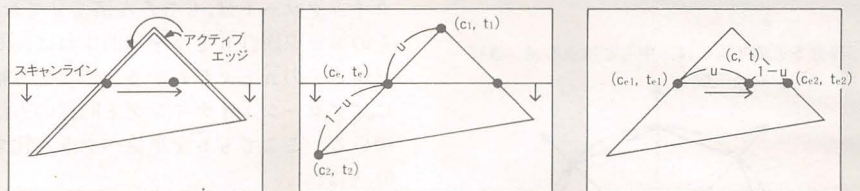
図5-5 フォンシェイディング

各頂点の法線を

滑らかにつないで

各点で色を計算する

グローシェイディングは平面にグラデーションをかけて曲面に見せていたが，フォンシェイディングでは平面を曲面に近似してから色を求めている

図5-6 頂点の属性を補間する(グローシェイディングの場合)

Zバッファアルゴリズムでは，スキャンラインごとにエッジとの交点を求め，その間をつないでいた。ここにシェイディング補間を組み込む

エッジの端点の属性を補間する。(色と透過率)

$$\begin{cases} c_e = (1-u)c_1 + uc_2 \\ t_e = (1-u)t_1 + ut_2 \end{cases}$$

この(c_e, t_e)をスキャンラインバッファに格納する

スキャンラインバッファの属性を補間する。

$$\begin{cases} c = (1-u)c_{e1} + uc_{e2} \\ t = (1-u)t_{e1} + ut_{e2} \end{cases}$$

これでピクセルごとの属性が求められた

読み終わったあとでまとめて行う。2)以降も同じ。

2) 引数に，

位置ベクトル$\vec{v}$：i番目の頂点$\vec{v}_i$

法線ベクトル$\vec{n}$：i番目の仮想法線$\vec{n}_i$

を代入して，レンダリング処理ルーチンを呼び出す。これで各項点の本体色c_iと透過率t_iが得られる。

3) エッジの始点と終点のcとtに，2)で求めたc_iとt_i，c_{i+1}とt_{i+1}を代入する。これでエッジの頂点属性の設定は終わり。

4) ポリゴンをスキャンコンバートしながら，頂点属性を補間していく。グローシェイディングでは，本体色cと透過率tを補間する。

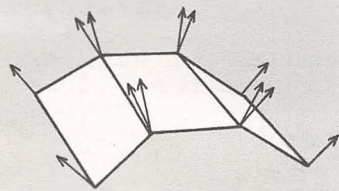
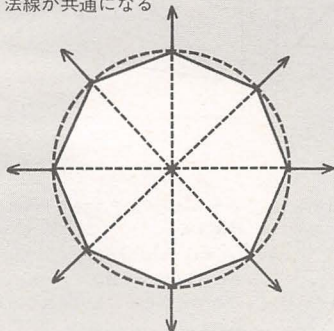
5) まず，メインループでは，エッジの始点と終点間で「頂点属性」のcとtを補間する。あるスキャンラインを処理しているとき，処理がアクティブエッジのどのくらいまで進んだかは，0～1の実数値で表すことができる。具体的には，y方向にループした回数を，エッジのy方向の大きさで割ってやれば求められる。この値をuと置く。始点の頂点属性がc_iとt_i，終点がc_2とt_2で表されていれば，エッジのスキャンラインとの交点での属性は，

$c_e=(1-u)c_1+uc_2$

$t_e=(1-u)t_1+ut_2$

これらがこのエッジの「エッジ属性」になる。スキャンラインバッファに格納する。

図5-7 モデリングのためのヒント

頂点ごとに仮想法線ベクトルを指定するのはグローシェイディングでもフォンシェイディングでも同じ。隣り合ったポリゴンの法線は共通にする。そうしないと滑らかにつながらない

球面を近似するには，中心と頂点を結ぶ直線と法線が共通になる

6) 次に，関数zbuff_line( )中のループでは，スキャンライン上で「エッジ属性」を補間して「ピクセル属性」を得る。スキャンラインバッファには，交点のエッジの情報が偶数個入っている（交点のx座標でのクイックソートは，もちろん済ませておく）。そのエッジ属性cとtを補間すれば，ピクセルごとのcとtがわかる。たとえば，エッジ属性が（c_{e1}，t_{e1}）と（c_2，t_2）のように格納されていたとする。x方向にどれだけ処理が進んだかを示す値も，同様にx方向の大きさでx方向のループ回数を割って求める。これもuと置くと，ピクセルの属性は，

$c=(1-u)c_{e1}+uc_{e2}$

$t=(1-u)t_{e1}+ut_{e2}$

ここでピクセルに書き込む処理を行う。

フォンシェイディング

1) ポリゴンの各頂点での頂点と仮想法線を読み込む。ここはグローシェイディングと同じ。

2) エッジの始点と終点の$\vec{n}$と$\vec{v}$に，1)で求めた$\vec{n}_i$と$\vec{v}_i$，$\vec{n}_{i+1}$と$\vec{v}_{i+1}$を代入する。これでエッジの頂点属性の設定は終わり。

3) ポリゴンをスキャンコンバートしながら，頂点属性を補間していく。フォンシェイディングでは，仮想法線$\vec{n}$と位置ベクトル$\vec{v}$を補間する。

4) まず，メインループでは，エッジの始点と終点間で「頂点属性」の$\vec{n}$と$\vec{v}$を補間する。それがこのエッジの「エッジ属性」になる。補間の具体的な方法はグローシェイディングと同様。出てきた$\vec{n}$と$\vec{v}$をスキャンラインバッファに格納する。補間したあとは，ベクトルの長さも変わるので，$\vec{n}$を単位ベクトル化するのを忘れずに。

5) 次に，関数zbuff_line( )中のループでは，スキャンライン上で「エッジ属性」を補間して「ピクセル属性」を得る。スキャンラインバッファには，交点のエッジの情報が偶数個入っている（交点のx座標でのクイックソートは，もちろん済ませておく）。そのエッジ属性$\vec{n}$と$\vec{v}$を補間すれば，ピクセルごとの$\vec{n}$と$\vec{v}$がわかる。ここでも補間にはグローシェイディングと同様の方法を用いる。ここでも$\vec{n}$を単位ベクトル化する。

6) 引数に，

位置ベクトル$\vec{v}$：補間後の位置ベクトル$\vec{v}$

法線ベクトル$\vec{n}$：補間後の仮想法線$\vec{n}$

を代入して，レンダリング処理ルーチンを呼び出す。ここで初めてポリゴンの本体色cと透過率tが得られる。ピクセルに書き込む処理を行う。

この3つの処理で，レンダリングを行う場所がそれぞれ違うのに注意しておこう。ここが処理速度やリアルさの度合いの違いになって現れている部分である。

モデリングのために

仮想法線のうまい設定の仕方

各項点に対して仮想的な法線を設定して曲面を表現するわけだが，ヒントを与えておこう（図5-7）。

円筒を角柱で近似したい場合など，隣り合った平面と平面の接合部には共通の辺が1本ある。この辺上の法線を2つの平面で共通にすると，仮想法線を補間したときも滑らかにつながる。したがってシェイディングも連続的になる。

球の各項点の仮想法線の方向は，球の中心と頂点とを結ぶ直線上にあるので，これを利用すれば球の仮想法線ベクトルは簡単に設定できる。

エッジの始点と終点の仮想法線を補間するので，この2つの仮想法線は180°反対方向を向いていてはいけない。

今後のテーマ

今回の紹介の範囲で，フォローしきれなかった表現について説明しておこう。これらはこれからの課題である。

光源関係

1) 複数光源

光源は1個と限定したが，実際のところいくつにでも増やせる。各光源について拡散反射強度とハイライト強度を計算し，単純にそれらの和をとればいいだけ。しかし，あとでいうシャドウ（影）の処理をさぼっているので，光源を複数にしただけでは，いたずらに計算時間が増えるだけで，有り難みの少ないものになってしまう。ひとつの物体に2つも3つも影ができて初めて複数光源のおいしさが出てくるのだから。

2) 異形光源（点，スポット，線，面，多面体光源など）

今回採用したのは点光源，しかも平行光線。いちばん単純である。リアルな効果を狙うなら，スポット光源（例：電気スタンド），線光源（蛍光灯），多角形や多面体光源（ちょっとおしゃれなビルでは，電球や蛍光灯をむき出しにしないで，四角いカバーのなかに収めているようなものが多い）など（図6-1）。あとのものほど配光特性が複雑で，やっぱり計算時間を喰うものになる。シャドウとも関連するが，線光源や多

面体光源などのように長さや面積を持った光源では，影が本影(影のなかでも暗い部分)と半影（薄暗い部分）に分かれる。これは論文級の課題といえよう。

周辺光のリアルな表現

周辺光（環境光）は，今回は一定としたのだが，シーン中の物体の配置の仕方や光源の数・形などでも微妙に変化させるのが理想的。ただ計算時間を考えると，かなり重い処理である。これまた論文クラスの研究テーマになり得る。

シャドウ：影

ある面に光源からの光が当たっているかどうかは，判定が単純ではない。だから長いこと影の表現は避けられてきたが，影がないとどうも物体が浮き上がって見え，不自然である（図6-2)。

少し面白いのが，次の事実。「光源の方向に仮に視点を置く。そこから不可視になる部分は，実際のシーンでの影になっている（わかるかな？）」だから，シャドウの実現には，

1） 光源から陰面除去を行う。
2） 本番の陰面除去の際に「光源からの隠面」を影の部分として処理する。

という2段階の処理を必要とする。どんなシャドウ表現アルゴリズムでも，この「2段階法」を踏襲しているようだ（ちなみにレイトレーシングでもそうなっている)。Zバッファアルゴリズムでこれを実現するには，1段階目で光源からシーンを作り，2段階目で本当のシーンを作る，そのどちらの処理にもZバッファアルゴリズムが使えるのだが，プログラムが複雑になるので今回はパスさせてもらった。

マッピング：手軽な表現力の強化

物体数を増やさずに複雑な表現をしたいときに，平面の上に色や凹凸などの情報，つまりテクスチャを疑似的に貼り付ければ済む場合がある。ありがちな例だが，ボールの表面に世界地図を貼り付ければ地球儀になる。さらに等高線から求められる凸凹を貼り付ければもっとリアルになる。エリアシング問題などもあって，滑らかに，自然にテクスチャを貼り付けるのは非常に面倒な処理。

アンチエリアシング：目に優しく

エリアシングとは，今回のサンプルを見るまでもなく，エッジがガタガタになってしまっている（ジャギー）現象など，一般にサンプリング解像度（平たくいえばドット数）の不足から，不自然なシーンが発生すること。いちばん気になるのはなんといってもエッジが滑らかに見えないこと。それを，滑らかに見せる処理がアンチエリアシング(エリアシング除去)。無限×無限ドットの，死ぬほど細かいスクリーンならアンチエリアシング処理なんて必要ないのだが，現実はそれほど甘くない。

通常のCGのアルゴリズムでは，エッジの形状はピクセルに色を書き込む直前まで正確に計算されている（たとえば三角形の一部だとかいうふうに)。したがって，ポリゴンのエッジ部に当たるピクセルのなかでポリゴンがどのくらいの面積を占めるかも完全に正確に計算できる。だからアンチエリアシングを行う場合も，面積比を用いて適当な方法でピクセルの色を混ぜればいいのだ。これは，ピクセルに書き込む前にエッジからジャギーを取り除く処理を行う意味で，「前置きフィルタリング」と呼ばれるアンチエリアシングの手法だ。

しかし，Zバッファアルゴリズムは，像空間（デバイス座標系の）アルゴリズムであるがために，前置きフィルタリングの処理が使いにくい（不可能ではないらしいが，かなり難しい)。というのも，透視変換を終えた時点で頂点の座標値はすでに整数。エッジの発生もすべて整数演算。というわけで，エッジ部の解析にはかなりの困難が伴うことになる。面積比なんてとてもとても。この点ではレイトレーシング法とて事情は同じである。

Zバッファ法でもレイトレーシング法でも，よく使われるアンチエリアシングの手法は「後置きフィルタリング」である。これはいわゆるオーバーサンプリングである。すなわち，解像度以上の数のピクセルについて色を計算し，適当な方法で平均化する手法である。たとえば，512×512ドット表示をしたいときは，1024×1024ドット分の色を計算し，ピクセルに書き込む段階で2×2＝4ドット分の色の平均を取る。これだけでもかなり効果的なアンチエリアシングを期待することができる。

試しに，Z'sSTAFFで2倍拡大，ぼかし，2分の1倍縮小という手順で処理を加えてみる実験をしてみよう。それだけでけっこうエッジがならされて滑らかになっていることだろう。これはオーバーサンプリング法を使ったアンチエリアシングと似ている。

アンチエリアシングは，マッピング，アニメーション，いろいろなCG技法を検討するときに決して避けて通れない。処理も複雑かつ長時間になるものだし，現在も多くのCG研究者たちがよりよいアルゴリズムを求めている。奥が相当に深いので今回は勘弁させてもらうが，リアルな表現のためにいつかは真面目に取り組むべき問題だと思う。

ファイルのフォーマット：拡張版

アトリビュートの部分を拡張している。前回のアッパーコンパチとなっている（どんなもんだい）が，いかんせんテキストエディタでないと書けないし，レイトレーシングに比べて物体の数が極端に多いので管理が厄介なことこのうえない。たとえば球体の表現にしても，レイトレではプリミティブ1個ですむが，Zバッファではポリゴンを100枚以上使わないと丸みが出ない。というわけだから，どうしても優れたCADの力を借りるか，データを自動生成するプログラムでも作らないことには，使い物にならない。まことに遺憾である。

図6-2 シャドウ（影）

影のない物体は浮き上がって見える

図6-1 異形光源

平行光源（今回採用）

点光源

スポット光源

線光源

面光源

多面体光源

このプログラムを走らせると，「主記憶が不足しています！」というメッセージをシステムが出してくることがある。RAM ディスクを載せていたりする場合には，このことで少しセコい計算をしてみる。

掲載したプログラムは，XC でコンパイルすると，ファイルのサイズが30Kバイト強になる。それから，いちばんメモリを喰っているのがエッジ構造体で，プリプロセッサには最大1024本と指定してある。で，size of (EDGE)×1024を計算すると 200K バイトを超える。次に大喰いなのがスキャンラインバッファで，100K近くにもなる（すべてのエッジがスキャンラインと交差しても処理できるように，要素を1024個まるまる取ってしまったが，これは取り過ぎであろう)。そのほかでも50Kくらい取っているから，フリーエリアが 400K バイトくらい必要な計算になる。こんなメモリ浪費の原因は，ベクトル構造体のサイズが大きいからだ。倍精度実数がx，y，zで３つである。これだけで24バイト。これが何百何千となく取られている。

それだけのメモリを使っていながら，表現できるのは，せいぜい256枚のポリゴン（定数MAXEDGEとMAXPOLYGONの関係は，１枚のポリゴンが四角形までとして，MAXEDGE＝MAXPOLYGON×4を目安にするとよい)。ちなみにサンプルの花瓶には，100枚以上のポリゴンを使っている。

これに対処するには，

1)　メモリを4Mバイトくらい拡張する(おおっお金持ち！)
2)　整数演算で済むように，プログラムを改造する。

これは2)の整数化のアプローチが現実的で，かつ高速化も図れそうだ。

高速化の話が出たので，速度のこともいっておこう。サンプル１の花瓶は，透明なものをフォンシェイディングで表示させると，256×256ドットの場合，数値演算プロセッサなしで15分，あっても９分くらいかかる。いくら教育的なサンプルプログラムといっても，ちょっとばかり遅い。

まだまだ道は遠い

以上，長々と説明してきたが，一応これで今回のＺバッファアルゴリズムについての紹介は終わる。こうした内容は本気で解説しようとすると本が１冊書けるほどの内容になってしまう。それを２カ月でこなしてしまおうというのだから，多少の無理はご理解いただきたい。多少，無駄話も多か

表１　シーン定義ファイルのフォーマット

ファイルネームは“????.SCN”

VIEW	X_{view} Y_{view} Z_{view}	視点（論理座標）
TARGET	X_{target} Y_{target} Z_{target}	注視点（論理座標）
ZOOM	α	画角（単位は度）
RATIO	ratio	ピクセルの縦横比（1.25が標準）
AREA	sx_1 sy_1 sx_2 sy_2	画面に表示する範囲（各0～511）
LIGHT	x_L y_L z_L	光源（平行光線）の方向
	R_L G_L B_L	光源の色（各０～１）
AMBIENT	R_a G_a B_a	周辺光の色（各０～１）
BACK	R_b G_b B_b	背景の色（各０～１）
END		

表２　物体定義ファイルのフォーマット

ファイルネームは“????.OBJ”

ATTRIBUTE		アトリビュート定義部
attr 1		アトリビュート名 ※３つのモードがあるが，そのうち必ずひとつ以上指定すること。たとえば，DIFFUSEを指定したら，自動的に拡散反射モードが設定される。指定しなかったモードの処理は行われない。
(DIFFUSE	R_d G_d B_d)	拡散反射係数（各０～１）
(SPECULAR	R_s G_s B_s	鏡面反射係数（各０～１）
	hl hp)	ハイライト定数（各１～）
(TRANSPARENT	R_t G_t B_t	透明度係数（各０～１）
	t_{min} t_{max}	最小，最大透過率（各０～１）
	tp)	透過率定数（１～）
END		アトリビュート定義の終わり
attr 2		
⋮		
END		
⋮		
attr N		
⋮		
END		
END		アトリビュート定義部の終わり
POLYGON		ポリゴン定義部
$poly_1$		ポリゴン名
ATTRIBUTE	a-name	アトリビュート
SHADING	constant	シェイディングモード
x_0 y_0 z_0		コンスタントシェイディングでは頂点の座標だけ指定する
x_1 y_1 z_1		
⋮		
x_n y_n z_n		
END		ポリゴン定義の終わり
$poly_2$		
ATTRIBUTE	a-name	
SHADING	Gouraud (Phong)	シェイディングモード グローシェイディングやフォンシェイディングでは頂点ごとの法線ベクトルも指定する
x_0 y_0 z_0	x_{n0} y_{n0} z_{n0}	
x_1 y_1 z_1	x_{n1} y_{n1} z_{n1}	
⋮		
x_N y_N z_N	x_{nN} y_{nN} z_{nN}	
END		
⋮		
$poly_n$		
⋮		
END		
END		ポリゴン定義部の終わり

ったし，この程度の解説の量では全部の分野をカバーしたとは僕自身も思っていない。

1980年代はコンピュータグラフィックのリアルな表現が飛躍的に進歩した時期である。いかにもCG然とした雰囲気の画像は，もう過去のものになりつつある。写真と見まがうばかりの表現もすぐそこにきている。しかるにこの2回でやったことは，表現力もスピードも，ポリゴンの数そのほかも，せいぜい'70年代後半のものである。逆にいえばX68000の力が当時の最先端の機械に追いついてきているということでもある。

これまで僕は，コンピュータグラフィックの発達の歴史を追うように説明してきたつもりだ。これから先も日々精進していきたいわけである。ということで，いつも「今後の課題」を残してばかりで終わってしまうというのが，イマイチ努力不足の感があり恐縮はしているのだが，夏休みの宿題と違って，ためておいてもあとで困るわけではないのが唯一救いだな（なんというワガママな発言）。

また，機会があれば，これらの宿題を少しずつでも整理していきたいと思っている。

参考文献

デビッドF.ロジャース，セイコー電子工業(株)電子機器事業部訳，山口富士夫監修，実践コンピュータグラフィックス，日刊工業新聞社，6,500円（税別）

リスト1 Zバッファアルゴリズム

```
=================  zbuff5.c  =================
  1: /*       Zバッファアルゴリズム                 */
  2: /*             レンダリング                    */
  3: /*        拡散反射、ハイライト、透明感          */
  4: /*        スムーズシェーディング                */
  5:
  6: /*   ※注釈のあるあたりが先月からの変更点        */
  7:
  8:
  9:
 10: #include  <stdio.h>
 11: #include  <stdlib.h>
 12: #include  <string.h>
 13: #include  <basic0.h>
 14: #include  <graph.h>
 15: #include  <math.h>
 16: #include  <time.h>
 17: #include  "sub.h"
 18:
 19:
 20: #define  MAX_EDGE  1024   /*メモリ容量やオブジェクトの複雑さ*/
 21: #define  MAX_POLYGON  256 /*   で適当に変えてね */
 22: #define  MAX_ATTRIBUTE  32
 23: #define  NAME_LEN  32
 24: #define  ON    1
 25: #define  OFF    0
 26: /*
 27: #define  Z_SCALE     32768
 28: #define  OK    1
 29: #define  WRONG    0
 30: #define  ACTIVE   1
 31: #define  INACTIVE 0
 32: #define  CONSTANT 0 /* コンスタントシェーディング */
 33: #define  GOURAUD  1 /* グロー (Gouraud) シェーディング */
 34: #define  PHONG    2 /* フォン (Phong) シェーディング */
 35: #define  FNAME_LEN  32
 36: #define  ARG_LEN    8
 37: #define  MAX_PIXEL  512
 38:
 39: typedef struct {
 40:   int  flag;
 41:   int  x, y, z;
 42:   int  dx, dy, dz;
 43:   int  sx, sz;
 44:   int  ex, ez;
 45:   int  ry;
 46:   /* エッジの端点での頂点属性：始点と終点の間で補間する */
 47:   /* グローシェーディング用に追加された頂点属性 */
 48:   VECTOR c1, c2; /*本体色（拡散反射成分とハイライト成分）*/
 49:   double t1, t2; /*透明度 */
 50:   /* フォンシェーディング用に追加された頂点属性 */
 51:   VECTOR  n1, n2; /* 仮想法線ベクトル */
 52:   VECTOR  v1, v2; /* 位置ベクトル */
 53: } EDGE;
 54: EDGE  edge[ MAX_EDGE ];
 55: int  n_edge;
 56:
 57:
 58: typedef struct {
 59:   char  name[ NAME_LEN ];
 60:   int  shading_type;
 61:   int  y_min, y_max;
 62:   int  edge1, edge2;
 63:   int  a_edge1, a_edge2;
 64:   int  attribute;
 65:   /* 削除：color */
 66:   /* コンスタントシェーディング用に追加された面の属性 */
 67:   VECTOR  c;       /* 本体色 */
 68:   double  t;       /* 透明度 */
 69: } POLYGON;
 70: POLYGON  poly[ MAX_POLYGON ];
 71: int  n_poly;
 72:
 73:
 74: typedef struct {
 75:   char  name[ NAME_LEN ];
 76:   int  diffuse;     /* 拡散反射モード：ON または OFF */
 77:   VECTOR  c_dif;    /* 拡散反射色 */
 78:   int  specular;    /* 鏡面反射(ハイライト)モード：ON または OFF */
 79:   VECTOR  c_spc;    /* RGBごとの反射率（ハイライトの色）*/
 80:   double  hl, hp;   /* ハイライト定数（強さ、収束率）*/
 81:   int  transparent;/* 透明感モード：ON または OFF */
 82:   VECTOR  c_trn;    /* RGBごとの透過率（透過光の色）*/
 83:   double  t_min, t_max, tp;  /* 透過率近似係数 */
 84: } ATTR;
 85: ATTR  attr[ MAX_ATTRIBUTE ];
 86: int  n_attr;
 87:
 88:
 89: typedef struct {
 90:   int  x, z;
 91:   int  st;       /* シェーディングの種類 */
 92:   /* 削除：color */
 93:   /* エッジのスキャンラインとの交点での属性 */
 94:   VECTOR  c; /* 本体色（グローシェーディング）*/
 95:   double  t; /* 透明度（グローシェーディング）*/
 96:   VECTOR  n; /* 仮想法線ベクトル（フォンシェーディング）*/
 97:   VECTOR  v; /* 位置ベクトル（フォンシェーディング）*/
 98: } SLBUF;
 99: SLBUF  slbuf[ MAX_EDGE ];
100:
101:
102: typedef  struct {
103:   int  depth;
104:   VECTOR  c; /* 各ピクセルの色を格納する（ベクトル表現）*/
105: } ZBUFF;
106: ZBUFF  zbuff[ MAX_PIXEL ];
107:
108:
109: VECTOR  view, target;
110: double  zoom, ratio;
111: int  s_x1, s_y1, s_x2, s_y2;
112: int  s_mx, s_my, s_dx, s_dy;
113: VECTOR  light, c_light, c_ambient, c_back;
114: MATRIX  rev;
115: double  d, dxy;
116: double  mag_x, mag_y, mag_z;
117: /* 削除：back */
118:
119:
120: int  read_data( char * );
121: int  compare_edge( EDGE *, EDGE * );
122: int  compare_slbuf( SLBUF *, SLBUF * );
123: void zbuff_line( int, SLBUF *, SLBUF *, ATTR * ); /*拡張*/
124: int  v_rgb( VECTOR * );
125: void pers( VECTOR *, int *, int *, int * );
126: void scanline_init( int );
127: void rendering( VECTOR *, double *, ATTR *, VECTOR *, VECTOR * );/* 拡張 */
128: VECTOR  *vector_inst( VECTOR *, VECTOR *, VECTOR *, double );    /* 追加 */
129: void  scanline_disp( int );              /* 追加 */
130:
131:
132: void  main( argc, argv )
133: int  argc;
134: char  *argv[];
135: {
136:   static int p0, p, y, e, i, s;
137:   static int start_time, end_time;
138:   static int sh_type, tr_mode;  /* シェーディングの種類と透明感モード */
139:   static double  u; /* 頂点属性の補間用パラメータ */
140:
141:   screen( 1, 3, 1, 1 );
142:
143:   if ( read_data( argv[1] )==WRONG ) return;
144:
145:   for ( p=0; p<n_poly; p++ ) {
146:     poly[p].a_edge1 = poly[p].edge1;
147:     poly[p].a_edge2 = poly[p].edge1-1;
148:   }
149:   time( &start_time );
150:   printf( "START: %s", ctime( &start_time ) );
151:
152:   for ( y = s_y1; y <= s_y2; y++ ) {
153:     scanline_init( y );
154:     /*各ポリゴンは2回ループする：*/
155:     /*1回目( p0 = 0 ～ n_poly-1 )は不透明ポリゴンだけを、*/
156:     /*2回目( p0 = n_poly ～ n_poly*2-1 )は(半)透明ポリゴンだけを処理する */
157:     for ( p0=0; p0<n_poly*2; p0++ ) {
158:       p = p0 % n_poly;
159:       tr_mode=( attr[poly[p].attribute].transparent==ON );
160:       if ( p0<n_poly && tr_mode || p0>=n_poly && !tr_mode ) continue;
161:       if ( poly[p].y_min>y || poly[p].y_max<y ) continue;
162:       sh_type = poly[p].shading_type;  /* シェーディングの種類 */
163:       for ( e = poly[p].a_edge2+1; e <= poly[p].edge2; e++ ) {
164:         if ( edge[e].y == y ) {
165:           edge[e].flag = ACTIVE;
166:           edge[e].ex   = -edge[e].dy;
167:           edge[e].ez   = -edge[e].dy;
168:           edge[e].ry   = edge[e].dy;
169:           poly[p].a_edge2 = e;
170:         }
171:         if ( edge[e].y > y ) break;
172:       }
173:       s=0;
174:       for ( e = poly[p].a_edge1; e <= poly[p].a_edge2; e++ ) {
175:         if ( edge[e].flag == INACTIVE ) continue;
176:         /* 頂点属性の補間用パラメータを計算する */
177:         if ( sh_type==GOURAUD || sh_type==PHONG )
178:           u = (double)(edge[e].dy - edge[e].ry) / (double)(edge[e].dy);
179:         if ( --edge[e].ry < 0 ) {
180:           edge[e].flag = INACTIVE;
181:           if ( e==poly[p].a_edge1 ) {
```

```
182:            while ( edge[e].flag == INACTIVE ) {
183:              e++;
184:              if ( e > poly[p].a_edge2 ) break;
185:            }
186:            poly[p].a_edge1=e;
187:            e--;
188:          }
189:          continue;
190:        }
191:        edge[e].ex += 2*edge[e].dx;
192:        slbuf[s].x = edge[e].x;
193:        while ( edge[e].ex >= 0 ) {
194:          edge[e].x += edge[e].sx;
195:          edge[e].ex -= 2*edge[e].dy;
196:        }
197:        edge[e].ez += 2*edge[e].dz;
198:        slbuf[s].z = edge[e].z;
199:        while ( edge[e].ez >= 0 ) {
200:          edge[e].z += edge[e].sz;
201:          edge[e].ez -= 2*edge[e].dy;
202:        }
203:        /* エッジ属性をスキャンラインバッファへ格納する */
204:        switch ( sh_type ) {
205:          /* コンスタントシェーディング: */
206:          /*   ポリゴン属性をそのままコピーする */
207:          case CONSTANT:
208:            vector_copy( &slbuf[s].c, &poly[p].c );
209:            if ( tr_mode ) slbuf[s].t = poly[p].t;
210:            break;
211:          /* グローシェーディング: */
212:          /*   頂点属性(本体色、透明度)を補間する */
213:          case GOURAUD:
214:            vector_inst( &slbuf[s].c, &edge[e].c1, &edge[e].c2, u );
215:            if ( tr_mode ) slbuf[s].t = (1.0-u)*edge[e].t1 + u*edge[e].t2;
216:            break;
217:          /* フォンシェーディング: */
218:          /* 頂点属性(仮想法線ベクトル、位置ベクトル)を補間する */
219:          case PHONG:
220:            vector_inst( &slbuf[s].n, &edge[e].n1, &edge[e].n2, u );
221:            vector_inst( &slbuf[s].v, &edge[e].v1, &edge[e].v2, u );
222:            unit( &slbuf[s].n, &slbuf[s].n );
223:            break;
224:        }
225:        slbuf[s].st = sh_type; /*シェーディングの種類をコピーする */
226:        s++;
227:      }
228:      if ( s>0 ) {
229:        qsort( slbuf, s, sizeof(SLBUF), compare_slbuf );
230:        for ( i=0; i<s-1; i+=2 )  /* 呼び出し方を拡張 */
231:          zbuff_line( y, &slbuf[i], &slbuf[i+1], &attr[poly[p].attribute] );
232:      }
233:    }
234:    scanline_disp( y );  /* 1スキャンラインの処理が終わったらまとめて表示 */
235:  }
236:  time( &end_time );
237:  printf( "END  : %s", ctime( &end_time ) );
238:
239:  return;
240: }
241:
242:
243: void  zbuff_line( y, s1, s2, a )  /* 引数を拡張 */
244: int  y;
245: SLBUF  *s1, *s2;
246: ATTR  *a;          /* アトリビュート情報を追加 */
247: {
248:   static  int  x1, z1, x2, z2, x, z;
249:   static  int  dx, dz, sx, sz, ez, i;
250:   static  int  tr_mode;  /* 透明感モード */
251:   /* エッジ属性の補間用パラメータ:次の違いに注意しよう */
252:   /* メインルーチン:頂点属性を補間→エッジ属性→スキャンラインバッファに格納 */
253:   /* 本ルーチン:スキャンラインバッファ中のエッジ属性を補間→ピクセル属性 */
254:   static  double  u;
255:   /* エッジ間で補間する属性(各ピクセルの属性になる)*/
256:   static  VECTOR  c, n, v;  /* 本体色、仮想法線ベクトル、位置ベクトル */
257:   static  double  t;     /* 透明度 */
258:
259:   x1=(s1->x);
260:   z1=(s1->z);
261:   x2=(s2->x);
262:   z2=(s2->z);
263:   dx=abs( x2-x1 );
264:   dz=abs( z2-z1 );
265:   sx=sgn( x2-x1 );
266:   sz=sgn( z2-z1 );
267:   ez=-dx;
268:   x=x1;
269:   z=z1;
270:   tr_mode=( a->transparent==ON );/* 透明感モード */
271:   for ( i=0; i<dx; i++ ) {  /* エッジの右端を処理しないのは透明感の処理で特に重要 */
272:     u = (double)i / (double)dx;  /* エッジ属性の補間用パラメータ */
273:     if ( x>s_x2 ) break;
274:     if ( x>=s_x1 ) {
275:       if ( z > zbuff[x].depth ) {
276:         /* エッジ属性を補間してピクセル属性を求める */
277:         switch ( s1->st ) {
278:           /* コンスタントシェーディング: */
279:           /*   エッジ属性をそのままコピー */
280:           case CONSTANT:
281:             vector_copy( &c, &s1->c );
282:             if ( tr_mode ) t = s1->t;
283:             break;
284:           /* グローシェーディング: */
285:           /*   エッジ属性を補間する */
286:           /*   補間した本体色、透明度を使う */
287:           case GOURAUD:
288:             vector_inst( &c, &s1->c, &s2->c, u );
289:             if ( tr_mode ) t = (1.0-u)*s1->t + u*s2->t;
290:             break;
291:           /* フォンシェーディング: */
292:           /*   エッジ属性を補間する */
293:           /*   補間した仮想法線ベクトル、位置ベクトルから */
294:           /*   この時点で初めて本体色、透明度を求める */
295:           case PHONG:
296:             vector_inst( &v, &s1->v, &s2->v, u );
297:             vector_inst( &n, &s1->n, &s2->n, u );
298:             unit( &n, &n );
299:             rendering( &c, &t, a, &n, &v );
300:             break;
301:         }
302:         if ( tr_mode ) {
303:           /* (半)透明ポリゴン: */
304:           /*   透過光に色をつける */
305:           v_star_v( &zbuff[x].c, &a->c_trn, &zbuff[x].c );
306:           /*   旧ピクセル色(透過光)と本体色を透明度を使って混合する */
307:           vector_inst( &zbuff[x].c, &c, &zbuff[x].c, t );
308:           /*   (半)透明体では、Zバッファの更新はしない */
309:         } else {
310:           /* 不透明ポリゴン: */
311:           /*   本体色をコピーする */
312:           vector_copy( &zbuff[x].c, &c );
313:           /*   Zバッファを更新する */
314:           zbuff[x].depth=z;
315:         }
316:       }
317:     }
318:     ez+=2*dz;
319:     while ( ez>=0 ) {
320:       z+=sz;
321:       ez-=2*dx;
322:     }
323:     x+=sx;
324:   }
325:   return;
326: }
327:
328:
329: void  scanline_init( y )
330: int  y;
331: {
332:   int  x;
333:
334:   /* Zバッファの構造が変わったので、初期化ルーチンは全面改訂 */
335:   for ( x=s_x1; x<=s_x2; x++ ) {
336:     vector_copy( &zbuff[x].c, &c_back );  /* 背景 */
337:     zbuff[x].depth=0;       /* Zバッファのデプスは0 */
338:   }
339:   return;
340: }
341:
342:
343: void  scanline_disp( y )   /* Zバッファの内容を表示する */
344: int  y;
345: {
346:   int  x;
347:
348:   for ( x=s_x1; x<=s_x2; x++ )
349:     pset( x, y, v_rgb( &zbuff[x].c ) );
350:   return;
351: }
352:
353:
354: /* アトリビュート(a)と法線ベクトル(n)と位置ベクトル(v)から、本体色(c)と透明度(t)を得る */
355: void  rendering( c, t, a, n, v ) /* 引数を拡張 */
356: VECTOR  *c, *n, *v;                  /* 位置ベクトルを追加 */
357: double  *t;                          /* 戻り値に透明度を追加 */
358: ATTR  *a;
359: {
360:   static  double  cos_t;
361:   static  double  cos_a;   /* ハイライト用:cosα */
362:   static  double  ns;      /* 透明度用:法線の視線方向成分 */
363:   static  VECTOR  s, r;    /* 視線ベクトル、反射ベクトル */
364:   static  VECTOR  cd, cs;  /* 拡散反射色、ハイライト色 */
365:
366:   /* 拡散反射 */
367:   cos_t=v_dot_v( n, &light );     /* 光線と法線の角度:θ */
368:   if ( cos_t<0.0 ) cos_t=0.0;     /* θ≧90°:影になっている */
369:   if ( a->diffuse == ON ) {
370:     v_dot_s( &cd, &c_light, cos_t );  /* 面の明るさはcosθに比例する */
371:     v_plus_v( &cd, &cd, &c_ambient ); /* 周辺光 */
372:     v_star_v( &cd, &cd, &a->c_dif );  /* この2つは拡散反射 */
373:   } else {
374:     set_vector( &cd, 0.0, 0.0, 0.0 ); /* 拡散反射なし */
375:   }
376:
377:   /* 鏡面反射(ハイライト) */
378:   v_minus_v( &s, &view, v );           /* 視線ベクトル */
379:   unit( &s, &s );
380:   if ( a->specular == ON && cos_t>0.0 ) {     /* 影の部分にハイライトはできない */
381:     v_dot_s( &r, n, 2.0*cos_t );      /* 反射ベクトル(大きさは1)*/
382:     v_minus_v( &r, &r, &light );
383:     cos_a=v_dot_v( &r, &s );      /* 視線と反射の角度はα */
384:     if ( cos_a<0.0 ) cos_a=0.0;      /* α≧90°:ハイライトなし */
385:     v_dot_s( &cs, &c_light, (a->hl)*pow( cos_a, a->hp ) );  /* ハイライト色計算
*/
386:     v_star_v( &cs, &cs, &a->c_spc );
387:   } else {
388:     set_vector( &cs, 0.0, 0.0, 0.0 );  /* ハイライトなし */
389:   }
390:
391:   /* シェーディング関数:拡散反射色とハイライト色→本体色 */
392:   v_plus_v( c, &cd, &cs );     /* cは戻り値 */
393:
394:   /* 透明感 */
395:   if ( a->transparent == ON ) {
396:     ns = fabs( v_dot_v( n, &s ) );     /* 法線の視線方向成分 */
397:     *t= a->t_min + ( a->t_max - a->t_min )*pow( ns, a->tp ); /* 透明度 */
398:   }              /* tも戻り値 */
399:   return;
400: }
401:
402:
403: int  read_data( filename )
404: char  *filename;
405: {
406:   static  FILE  *f_scn, *f_obj;
```

▶満開の電子ちゃんは，やはり裏表紙向きだと思うのですが……。
升井　晋也（20）福岡県

```
407:    static char fname_scn[ FNAME_LEN ], fname_obj[ FNAME_LEN ];
408:    static char arg_s[ ARG_LEN ];
409:    static int arg_i1, arg_i2, arg_i3, arg_i4;
410:    static int err=0, stop, i;
411:    static int sh_type;    /* シェーディングの種類 */
412:    static int x0, y0, z0, x1, y1, z1, x2, y2, z2, x, y, z;
413:    static int a, b;
414:    static double u;       /* クリッピング時の頂点属性補間 */
415:    static double cos_t, sin_t, cos_p, sin_p;
416:    static VECTOR r;
417:    static VECTOR s;
418:    /* 削除:c */
419:    static VECTOR na;
420:    static VECTOR v;
421:    static VECTOR va0, va1, va2;
422:    static VECTOR n, n0, n1, n2; /* 頂点の仮想法線ベクトル */
423:    static VECTOR v0, v1, v2;    /* 頂点の位置ベクトル */
424:
425:    strcpy( fname_scn, filename );
426:    strcat( fname_scn, ".SCN" );
427:    if ( ( f_scn=fopen( fname_scn, "rt" ) )==NULL ) {
428:      printf( "ERROR: file %s doesn't exist.¥n", fname_scn );
429:      return( WRONG );
430:    }
431:    for (;;) {
432:      fscanf( f_scn, "%s", arg_s );
433:      if ( strcmpi( arg_s, "VIEW" )==0 ) {
434:        vector_read( f_scn, &view );
435:        continue;
436:      }
437:      if ( strcmpi( arg_s, "TARGET" )==0 ) {
438:        vector_read( f_scn, &target );
439:        continue;
440:      }
441:      if ( strcmpi( arg_s, "ZOOM" )==0 ) {
442:        fscanf( f_scn, "%F", &zoom );
443:        zoom *= 0.0174532;
444:        continue;
445:      }
446:      if ( strcmpi( arg_s, "RATIO" )==0 ) {
447:        fscanf( f_scn, "%F", &ratio );
448:        continue;
449:      }
450:      if ( strcmpi( arg_s, "AREA" )==0 ) {
451:        fscanf( f_scn, "%d %d %d %d", &s_x1, &s_y1, &s_x2, &s_y2 );
452:        if ( s_x1>s_x2 ) swap( &s_x1, &s_x2 );
453:        if ( s_y1>s_y2 ) swap( &s_y1, &s_y2 );
454:        s_mx = ( s_x1 + s_x2 )/2;
455:        s_my = ( s_y1 + s_y2 )/2;
456:        s_dx = s_x2 - s_x1 +1;
457:        s_dy = s_y2 - s_y1 +1;
458:        continue;
459:      }
460:      if ( strcmpi( arg_s, "LIGHT" )==0 ) {
461:        vector_read( f_scn, &light );
462:        unit( &light, &light );
463:        vector_read( f_scn, &c_light );
464:        continue;
465:      }
466:      if ( strcmpi( arg_s, "AMBIENT" )==0 ) {
467:        vector_read( f_scn, &c_ambient );
468:        continue;
469:      }
470:      if ( strcmpi( arg_s, "BACK" )==0 ) {
471:        vector_read( f_scn, &c_back );
472:        /* 削除:backの計算 */
473:        continue;
474:      }
475:      if ( strcmpi( arg_s, "END" )==0 ) break;
476:      printf( "ERROR in %s: illegal command; %s¥n", fname_scn, arg_s );
477:      err=1;
478:      break;
479:    }
480:    fclose( f_scn );
481:    if ( err==1 ) return( WRONG );
482:
483:    v_minus_v( &r, &view, &target );
484:    dxy=sqrt( r.v[0]*r.v[0] + r.v[1]*r.v[1] );
485:    d=sqrt( r.v[0]*r.v[0] + r.v[1]*r.v[1] + r.v[2]*r.v[2] );
486:    if ( dxy!=0.0 ) {
487:              cos_t=r.v[0]/dxy; sin_t=r.v[1]/dxy;
488:              cos_p=dxy/d;      sin_p=r.v[2]/d;
489:              set_matrix( &rev,
490:                          -sin_t, cos_t, 0.0,
491:         -sin_p*cos_t, -sin_p*sin_t, cos_p,
492:          cos_p*cos_t,  cos_p*sin_t, sin_p );
493:    } else {
494:      if ( r.v[2]>0.0 )
495:        set_matrix( &rev,
496:           0.0, 1.0, 0.0,
497:          -1.0, 0.0, 0.0,
498:           0.0, 0.0, 1.0 );
499:      else
500:        set_matrix( &rev,
501:            0.0, 1.0,  0.0,
502:            1.0, 0.0,  0.0,
503:            0.0, 0.0, -1.0 );
504:    }
505:    mag_x=(double)s_dx / (2.0*d*tan( zoom/2.0 ));
506:    mag_y=-mag_x*ratio;
507:    mag_z=(double)Z_SCALE;
508:
509:    strcpy( fname_obj, filename );
510:    strcat( fname_obj, ".OBJ" );
511:    if ( ( f_obj=fopen( fname_obj, "rt" ) )==NULL ) {
512:      printf( "ERROR: file %s doesn't exist.¥n", fname_obj );
513:      return( WRONG );
514:    }
515:
516:    n_attr=0;
517:    fscanf( f_obj, "%s", arg_s );
518:    if ( strcmpi( arg_s, "ATTRIBUTE" )!=0 ) {
519:      err=1;
520:      printf( "ERROR in %s: begin without 'attribute'command; %s¥n", fname_obj,
arg_s );
521:      fclose( f_obj );
522:      return( WRONG );
523:    }
524:    for (;;) {
525:      fscanf( f_obj, "%s", arg_s );
526:      if ( strcmpi( arg_s, "END" )==0 || err==1 ) break;
527:      strcpy( attr[n_attr].name, arg_s );
528:
529:      attr[n_attr].diffuse=OFF;
530:      attr[n_attr].specular=OFF;    /* 追加 */
531:      attr[n_attr].transparent=OFF; /* 追加 */
532:      for (;;) {
533:        fscanf( f_obj, "%s", arg_s );
534:        if ( strcmpi( arg_s, "DIFFUSE" )==0 ) {
535:          attr[n_attr].diffuse=ON;
536:          vector_read( f_obj, &attr[n_attr].c_dif );
537:          continue;
538:        }
539:        /* ここから追加 */
540:        if ( strcmpi( arg_s, "SPECULAR" )==0 ) {
541:          attr[n_attr].specular=ON;
542:          vector_read( f_obj, &attr[n_attr].c_spc );
543:          fscanf( f_obj, "%F %F", &attr[n_attr].hl, &attr[n_attr].hp );
544:          continue;
545:        }
546:        if ( strcmpi( arg_s, "TRANSPARENT" )==0 ) {
547:          attr[n_attr].transparent=ON;
548:          vector_read( f_obj, &attr[n_attr].c_trn );
549:          fscanf( f_obj, "%F %F %F", &attr[n_attr].t_min, &attr[n_attr].t_max,
&attr[n_attr].tp );
550:          continue;
551:        }
552:        /* ここまで */
553:        if ( strcmpi( arg_s, "END" )==0 ) break;
554:        printf( "ERROR in %s: illegal attribute; %s¥n", fname_obj, arg_s );
555:        err=1;
556:        break;
557:      }
558:      n_attr++;
559:    }
560:    if ( err==1 ) {
561:      fclose( f_obj );
562:      return( WRONG );
563:    }
564:
565:    n_poly=0;
566:    n_edge=0;
567:
568:    fscanf( f_obj, "%s", arg_s );
569:    if ( strcmpi( arg_s, "POLYGON" )!=0 ) {
570:      err=1;
571:      printf( "ERROR in %s: 'polygon'command is needed; %s¥n", fname_obj, arg_s
);
572:      fclose( f_obj );
573:      return( WRONG );
574:    }
575:    for (;;) {
576:      fscanf( f_obj, "%s", arg_s );
577:      if ( strcmpi( arg_s, "END" )==0 ) break;
578:      strcpy( poly[n_poly].name, arg_s );
579:
580:      fscanf( f_obj, "%s", arg_s );
581:      if ( strcmpi( arg_s, "ATTRIBUTE" )!=0 ) {
582:        printf( "ERROR in %s: 'attribute'command is needed; %s¥n", fname_obj, a
rg_s );
583:        err=1;
584:        break;
585:      }
586:      fscanf( f_obj, "%s", arg_s );
587:      i=0;
588:      for (;;) {
589:        if ( strcmpi( arg_s, attr[i].name )==0 ) {
590:          poly[n_poly].attribute=i;
591:          break;
592:        }
593:        i++;
594:        if ( i==n_attr ) {
595:          printf( "ERROR in %s: this attribute-name doesn't exist; %s¥n", fname
_obj, arg_s );
596:          err=1;
597:          break;
598:        }
599:      }
600:      if ( err==1 ) break;
601:
602:      fscanf( f_obj, "%s", arg_s );
603:      if ( strcmpi( arg_s, "SHADING" )!=0 ) {
604:        printf( "ERROR in %s: 'shading'command is needed; %s¥n", fname_obj, arg
_s );
605:        err=1;
606:        break;
607:      }
608:      fscanf( f_obj, "%s", arg_s );
609:      for (;;) {
610:        if ( strcmpi( arg_s, "CONSTANT" )==0 ) {
611:          poly[n_poly].shading_type=CONSTANT;
612:          break;
613:        }
614:        /* ここから追加 */
615:        if ( strcmpi( arg_s, "GOURAUD" )==0 ) {
616:          poly[n_poly].shading_type=GOURAUD;
617:          break;
618:        }
619:        if ( strcmpi( arg_s, "PHONG" )==0 ) {
620:          poly[n_poly].shading_type=PHONG;
621:          break;
622:        }
623:        /* ここまで */
624:        err=1;
625:        printf( "ERROR in %s: bad shading type; %s¥n", fname_obj, arg_s );
626:        break;
```

▶ショートプログラムを連載するということですが，大変うれしく思います。いまから8月号が楽しみです。ちなみに小説も短編が好きです（関係ないけど）。

幸　久稔（27）鹿児島県

```
627:     }
628:     if ( err==1 ) break;
629:
630:     poly[n_poly].edge1=n_edge;
631:     stop=0;
632:     sh_type = poly[n_poly].shading_type;   /* シェーディングの種類 */
633:
634:     vector_read( f_obj, &v );
635:     vector_copy( &v0, &v );          /* 第1頂点の位置ベクトル */
636:     vector_copy( &va0, &v0 );
637:     pers( &v, &x, &y, &z );
638:     x0=x; y0=y; z0=z;
639:     /* スムーズシェーディングでは仮想法線ベクトルも読み込む */
640:     if ( sh_type==GOURAUD || sh_type==PHONG ) {
641:       vector_read( f_obj, &n );
642:       unit( &n, &n );
643:       vector_copy( &n0, &n );
644:     }
645:     poly[n_poly].y_min = poly[n_poly].y_max = y;
646:     for ( i=1;; i++ ) {
647:       x1=x; y1=y; z1=z;
648:     /* スムーズシェーディング：仮想法線、位置ベクトルも処理 */
649:       if ( sh_type==GOURAUD || sh_type==PHONG ) {
650:         vector_copy( &v1, &v );
651:         vector_copy( &n1, &n );
652:       }
653:       if ( vector_read( f_obj, &v )==OK ) {
654:         pers( &v, &x, &y, &z );
655:         if ( i==1 ) {
656:           vector_copy( &va1, &v );
657:         }
658:         if ( i==2 ) {
659:           vector_copy( &va2, &v );
660:           v_minus_v( &va1, &va1, &va0 );
661:           v_minus_v( &va2, &va2, &va0 ); /* 3頂点から2辺を得て */
662:           v_cross_v( &na, &va1, &va2 );  /* 真の法線を外積を使って求める */
663:           unit( &na, &na );
664:           v_minus_v( &s, &view, &va0 );
665:           if ( v_dot_v( &s, &na )<0.0 )
666:             v_dot_s( &na, &na, -1.0 );
667:           /* コンスタントシェーディング： */
668:           /*   ここでポリゴン属性（本体色、透明度）を決める */
669:           if ( sh_type==CONSTANT )
670:             rendering( &poly[n_poly].c, &poly[n_poly].t, &attr[poly[n_poly].a
ttribute], &na, &va0 );
671:               /* 削除：colorの計算 */
672:         }
673:         /* スムーズシェーディング：仮想法線ベクトルも読み込む */
674:         if ( sh_type==GOURAUD || sh_type==PHONG ) {
675:           vector_read( f_obj, &n );
676:           unit( &n, &n );
677:         }
678:       } else {
679:         fscanf( f_obj, "%s", arg_s );
680:         stop=1;
681:         x=x0; y=y0; z=z0;
682:         /* スムーズシェーディング：仮想法線、位置ベクトルも処理 */
683:         if ( sh_type==GOURAUD || sh_type==PHONG ) {
684:           vector_copy( &v, &v0 );
685:           vector_copy( &n, &n0 );
686:         }
687:       }
688:       x2=x; y2=y; z2=z;
689:       /* スムーズシェーディング：仮想法線、位置ベクトルも処理 */
690:       if ( sh_type==GOURAUD || sh_type==PHONG ) {
691:         vector_copy( &v2, &v );
692:         vector_copy( &n2, &n );
693:       }
694:       if ( y1>y2 ) {
695:         swap( &x1, &x2 );
696:         swap( &y1, &y2 );
697:         swap( &z1, &z2 );
698:         /* スムーズシェーディング：仮想法線、位置ベクトルも処理 */
699:         if ( sh_type==GOURAUD || sh_type==PHONG ) {
700:           vector_swap( &v1, &v2 );
701:           vector_swap( &n1, &n2 );
702:         }
703:       }
704:       if ( y1==y2 || y2<s_y1 || y1>s_y2 ) {
705:         if ( stop==1 ) break;
706:         continue;
707:       }
708:       if ( y1<s_y1 ) {
709:         a = s_y1 - y1;
710:         b = y2 - y1;
711:         x1 = x1+(x2-x1)*a/b;
712:         y1 = s_y1;
713:         z1 = z1+(z2-z1)*a/b;
714:         /* スムーズシェーディング：*/
715:         /*   クリップ後の新端点の仮想法線、位置ベクトルは補間して求める */
716:         if ( sh_type==GOURAUD || sh_type==PHONG ) {
717:           u = (double)a / (double)b;
718:           vector_inst( &v1, &v1, &v2, u );
719:           vector_inst( &n1, &n1, &n2, u );
720:           unit( &n1, &n1 );
721:         }
722:       }
723:       if ( y2>s_y2+1 ) {
724:         a = s_y2+1 - y1;
725:         b = y2 - y1;
726:         x2 = x1+(x2-x1)*a/b;
727:         y2 = s_y2+1;
728:         z2 = z1+(z2-z1)*a/b;
729:         /* スムーズシェーディング：*/
730:         /*   クリップ後の新端点の仮想法線、位置ベクトルは補間して求める */
731:         if ( sh_type==GOURAUD || sh_type==PHONG ) {
732:           u = (double)a / (double)b;
733:           vector_inst( &v2, &v1, &v2, u );
734:           vector_inst( &n2, &n1, &n2, u );
735:           unit( &n2, &n2 );
736:         }
737:       }
738:       edge[n_edge].flag=INACTIVE;
739:       edge[n_edge].x=x1;
740:       edge[n_edge].y=y1;
741:       edge[n_edge].z=z1;
742:       edge[n_edge].dx=abs( x2-x1 );
743:       edge[n_edge].dy=abs( y2-y1 );
744:       edge[n_edge].dz=abs( z2-z1 );
745:       edge[n_edge].sx=sgn( x2-x1 );
746:       edge[n_edge].sz=sgn( z2-z1 );
747:       /* スムーズシェーディング：仮想法線、位置ベクトルも処理 */
748:       if ( sh_type==GOURAUD || sh_type==PHONG ) {
749:         vector_copy( &edge[n_edge].v1, &v1 );
750:         vector_copy( &edge[n_edge].v2, &v2 );
751:         vector_copy( &edge[n_edge].n1, &n1 );
752:         vector_copy( &edge[n_edge].n2, &n2 );
753:       }
754:       if ( poly[n_poly].y_min > y1 ) poly[n_poly].y_min = y1;
755:       if ( poly[n_poly].y_max < y2 ) poly[n_poly].y_max = y2;
756:       n_edge++;
757:       if ( stop==1 ) break;
758:     }
759:     if ( err==1 ) break;
760:     poly[n_poly].edge2=n_edge-1;
761:     /* スムーズシェーディング： */
762:     /*   仮想法線が真の法線から見て裏向きならば、仮想法線を逆向きにする */
763:     if ( sh_type==GOURAUD || sh_type==PHONG ) {
764:       for ( i=poly[n_poly].edge1; i<=poly[n_poly].edge2; i++ ) {
765:         if ( v_dot_v( &na, &edge[i].n1 ) <0.0 )
766:           v_dot_s( &edge[i].n1, &edge[i].n1, -1.0 );
767:         if ( v_dot_v( &na, &edge[i].n2 ) <0.0 )
768:           v_dot_s( &edge[i].n2, &edge[i].n2, -1.0 );
769:         /* グローシェーディング： */
770:         /*   ここで頂点属性（本体色、透明度）を求める */
771:         if ( sh_type==GOURAUD ) {
772:           rendering( &edge[i].c1, &edge[i].t1, &attr[poly[n_poly].attribute],
&edge[i].n1, &edge[i].v1 );
773:           rendering( &edge[i].c2, &edge[i].t2, &attr[poly[n_poly].attribute],
&edge[i].n2, &edge[i].v2 );
774:         }
775:       }
776:     }
777:     qsort( &edge[ poly[n_poly].edge1 ], poly[n_poly].edge2 - poly[n_poly].edg
e1 + 1, sizeof(EDGE), compare_edge );
778:     n_poly++;
779:   }
780:   fclose( f_obj );
781:   if ( err==1 ) return( WRONG );
782:   return( OK );
783: }
784:
785:
786: int v_rgb( c )
787: VECTOR  *c;
788: {
789:   static  int  r, g, b;
790:
791:   r = (int)( c->v[0]*32.0 ); if ( r>31 ) r=31;
792:   g = (int)( c->v[1]*32.0 ); if ( g>31 ) g=31;
793:   b = (int)( c->v[2]*32.0 ); if ( b>31 ) b=31;
794:
795:   return( g<<11 | r<<6 | b<<1 );
796: }
797:
798:
799: void  pers( V, x, y, z )
800: VECTOR  *V;
801: int  *x, *y, *z;
802: {
803:   static  VECTOR  V0, V1;
804:   static  double  x1, y1, z1, mag;
805:
806:   v_minus_v( &V0, V, &target );
807:   m_dot_v( &V1, &rev, &V0 );
808:   x1 = V1.v[0];
809:   y1 = V1.v[1];
810:   z1 = V1.v[2];
811:    mag = 1.0/(d-z1);                              /* 変更 */
812:    *x = (int)(x1*d*mag*mag_x) + s_mx;            /* 変更 */
813:    *y = (int)(y1*d*mag*mag_y) + s_my;            /* 変更 */
814:    *z = (int)(mag*mag_z);                        /* 変更 */
815:
816:   return;
817: }
818:
819:
820: int  compare_edge( e1, e2 )
821: EDGE  *e1, *e2;
822: {
823:   if ( e1->y == e2->y )
824:     return( e1->dy - e2->dy );
825:   return( e1->y - e2->y );
826: }
827:
828:
829: int  compare_slbuf( s1, s2 )
830: SLBUF  *s1, *s2;
831: {
832:   return( s1->x - s2->x );
833: }
834:
835:
836: VECTOR  *vector_inst( V, V1, V2, u )    /* ベクトルの内挿（補間）*/
837: VECTOR  *V, *V1, *V2;
838: double  u;
839: {
840:   static  double  u1;
841:   static  int  i;
842:
843:   u1=1.0-u;
844:   for ( i=0; i<3; i++ )
845:     V->v[i] = u1*V1->v[i] + u*V2->v[i];
846:
847:   return( V );
848: }
```

▶アナログジョイスティック買うぞーっ！　金がないぞー！　高3だぞー！　おまけに学力がないぞー！　ぐっすん。
森下　寛和（17）鳥取県

リスト2 サンプルリスト

a) BL_P. SCN

```
=================  bl_p.SCN  =================
   1: view     200 -100 100
   2: target   0 0 0
   3: zoom     45
   4: ratio    1.25
   5: area     128 128 255 255
   6:
   7: light    1.0 1.0 1.0
   8:          0.9 0.9 0.9
   9: ambient  0.2 0.2 0.2
  10: back     0.5 1.0 1.0
  11:
  12: end
```

b) BL_P. OBJ

```
===============  BL_P.OBJ  ===============
   1: attribute
   2:   red  diffuse  1.0 0.0 0.0
   3:   specular  1.0 1.0 1.0  4 100
   4:   end
   5: end
   6:
   7: polygon
   8:   ball  attribute  red
   9:   shading  Phong
  10:   100     0     0  100     0     0
  11:    87    43    24   87    43    24
  12:    87    50     0   87    50     0
  13:    end
  14:
  15:   ball  attribute  red
  16:   shading  Phong
  17:   100     0     0  100     0     0
  18:    87    25    43   87    25    43
  19:    87    43    24   87    43    24
  20:    end
  21:
  22:   ball  attribute  red
  23:   shading  Phong
  24:   100     0     0  100     0     0
  25:    87     0    50   87     0    50
  26:    87    25    43   87    25    43
  27:    end
  28:
  29:   ball  attribute  red
  30:   shading  Phong
  31:   100     0     0  100     0     0
  32:    87   -24    43   87   -24    43
  33:    87     0    50   87     0    50
  34:    end
  35:
  36:   ball  attribute  red
  37:   shading  Phong
  38:   100     0     0  100     0     0
  39:    87   -43    25   87   -43    25
  40:    87   -24    43   87   -24    43
  41:    end
  42:
  43:   ball  attribute  red
  44:   shading  Phong
  45:   100     0     0  100     0     0
  46:    87   -50     0   87   -50     0
  47:    87   -43    25   87   -43    25
  48:    end
  49:
  50:   ball  attribute  red
  51:   shading  Phong
  52:   100     0     0  100     0     0
  53:    87   -43   -24   87   -43   -24
  54:    87   -50     0   87   -50     0
  55:    end
  56:
  57:   ball  attribute  red
  58:   shading  Phong
  59:   100     0     0  100     0     0
  60:    87   -25   -43   87   -25   -43
  61:    87   -43   -24   87   -43   -24
  62:    end
  63:
  64:   ball  attribute  red
  65:   shading  Phong
  66:   100     0     0  100     0     0
  67:    87     0   -50   87     0   -50
  68:    87   -25   -43   87   -25   -43
  69:    end
  70:
  71:   ball  attribute  red
  72:   shading  Phong
  73:   100     0     0  100     0     0
  74:    87    24   -43   87    24   -43
  75:    87     0   -50   87     0   -50
  76:    end
  77:
  78:   ball  attribute  red
  79:   shading  Phong
  80:   100     0     0  100     0     0
  81:    87    43   -25   87    43   -25
  82:    87    24   -43   87    24   -43
  83:    end
  84:
  85:   ball  attribute  red
  86:   shading  Phong
  87:   100     0     0  100     0     0
  88:    87    50     0   87    50     0
  89:    87    43   -25   87    43   -25
  90:    end
  91:
  92:   ball  attribute  red
  93:   shading  Phong
  94:    87    50     0   87    50     0
  95:    87    43    24   87    43    24
  96:    50    75    43   50    75    43
  97:    50    87     0   50    87     0
  98:    end
  99:
 100:   ball  attribute  red
 101:   shading  Phong
 102:    87    43    24   87    43    24
 103:    87    25    43   87    25    43
 104:    50    43    75   50    43    75
 105:    50    75    43   50    75    43
 106:    end
 107:
 108:   ball  attribute  red
 109:   shading  Phong
 110:    87    25    43   87    25    43
 111:    87     0    50   87     0    50
 112:    50     0    87   50     0    87
 113:    50    43    75   50    43    75
 114:    end
 115:
 116:   ball  attribute  red
 117:   shading  Phong
 118:    87     0    50   87     0    50
 119:    87   -24    43   87   -24    43
 120:    50   -43    75   50   -43    75
 121:    50     0    87   50     0    87
 122:    end
 123:
 124:   ball  attribute  red
 125:   shading  Phong
 126:    87   -24    43   87   -24    43
 127:    87   -43    25   87   -43    25
 128:    50   -75    43   50   -75    43
 129:    50   -43    75   50   -43    75
 130:    end
 131:
 132:   ball  attribute  red
 133:   shading  Phong
 134:    87   -43    25   87   -43    25
 135:    87   -50     0   87   -50     0
 136:    50   -87     0   50   -87     0
 137:    50   -75    43   50   -75    43
 138:    end
 139:
 140:   ball  attribute  red
 141:   shading  Phong
 142:    87   -50     0   87   -50     0
 143:    87   -43   -24   87   -43   -24
 144:    50   -75   -43   50   -75   -43
 145:    50   -87     0   50   -87     0
 146:    end
 147:
 148:   ball  attribute  red
 149:   shading  Phong
 150:    87   -43   -24   87   -43   -24
 151:    87   -25   -43   87   -25   -43
 152:    50   -43   -75   50   -43   -75
 153:    50   -75   -43   50   -75   -43
 154:    end
 155:
 156:   ball  attribute  red
 157:   shading  Phong
 158:    87   -25   -43   87   -25   -43
 159:    87     0   -50   87     0   -50
 160:    50     0   -87   50     0   -87
 161:    50   -43   -75   50   -43   -75
 162:    end
 163:
 164:   ball  attribute  red
 165:   shading  Phong
 166:    87     0   -50   87     0   -50
 167:    87    24   -43   87    24   -43
 168:    50    43   -75   50    43   -75
 169:    50     0   -87   50     0   -87
 170:    end
 171:
 172:   ball  attribute  red
 173:   shading  Phong
 174:    87    24   -43   87    24   -43
 175:    87    43   -25   87    43   -25
 176:    50    75   -43   50    75   -43
 177:    50    43   -75   50    43   -75
 178:    end
 179:
 180:   ball  attribute  red
 181:   shading  Phong
 182:    87    43   -25   87    43   -25
 183:    87    50     0   87    50     0
 184:    50    87     0   50    87     0
 185:    50    75   -43   50    75   -43
 186:    end
 187:
 188:   ball  attribute  red
 189:   shading  Phong
 190:    50    87     0   50    87     0
 191:    50    75    43   50    75    43
 192:     0    86    49    0    86    49
 193:     0   100     0    0   100     0
 194:    end
 195:
 196:   ball  attribute  red
 197:   shading  Phong
 198:    50    75    43   50    75    43
 199:    50    43    75   50    43    75
 200:     0    50    86    0    50    86
 201:     0    86    49    0    86    49
 202:    end
 203:
 204:   ball  attribute  red
 205:   shading  Phong
 206:    50    43    75   50    43    75
 207:    50     0    87   50     0    87
 208:     0     0   100    0     0   100
 209:     0    50    86    0    50    86
 210:    end
 211:
 212:   ball  attribute  red
 213:   shading  Phong
 214:    50     0    87   50     0    87
 215:    50   -43    75   50   -43    75
 216:     0   -49    86 0   -49    86
 217:     0     0   100 0     0   100
 218:    end
 219:
 220:   ball  attribute  red
 221:   shading  Phong
 222:    50   -43    75   50   -43    75
 223:    50   -75    43   50   -75    43
 224:     0   -86    50    0   -86    50
 225:     0   -49    86    0   -49    86
 226:    end
 227:
 228:   ball  attribute  red
 229:   shading  Phong
 230:    50   -75    43   50   -75    43
 231:    50   -87     0   50   -87     0
 232:     0  -100     0    0  -100     0
 233:     0   -86    50    0   -86    50
 234:    end
 235:
 236:   ball  attribute  red
 237:   shading  Phong
 238:    50   -87     0   50   -87     0
 239:    50   -75   -43   50   -75   -43
 240:     0   -86   -49    0   -86   -49
 241:     0  -100     0    0  -100     0
 242:    end
 243:
 244:   ball  attribute  red
 245:   shading  Phong
 246:    50   -75   -43   50   -75   -43
 247:    50   -43   -75   50   -43   -75
 248:     0   -50   -86    0   -50   -86
 249:     0   -86   -49    0   -86   -49
 250:    end
 251:
 252:   ball  attribute  red
 253:   shading  Phong
 254:    50   -43   -75   50   -43   -75
 255:    50     0   -87   50     0   -87
 256:     0     0  -100    0     0  -100
 257:     0   -50   -86    0   -50   -86
 258:    end
 259:
 260:   ball  attribute  red
 261:   shading  Phong
 262:    50     0   -87   50     0   -87
 263:    50    43   -75   50    43   -75
 264:     0    49   -86    0    49   -86
 265:     0     0  -100    0     0  -100
 266:    end
 267:
 268:   ball  attribute  red
 269:   shading  Phong
 270:    50    43   -75   50    43   -75
 271:    50    75   -43   50    75   -43
 272:     0    86   -50    0    86   -50
 273:     0    49   -86    0    49   -86
 274:    end
 275:
 276:   ball  attribute  red
 277:   shading  Phong
 278:    50    75   -43   50    75   -43
 279:    50    87     0   50    87     0
 280:     0   100     0    0   100     0
 281:     0    86   -50    0    86   -50
 282:    end
 283:
 284: end
```

▶なんなんだー，7月号の特集は。Cのリストがいっぱいだー！ XCってそんなに売れているの？ X68000がほしくなってしまうではないか！　浅野 克博 (16) 宮城県

新アルゴリズムの採用により高速化を実現

サイクロンExpress

Tan Akihiko
丹 明彦

X68000用レイトレーシングツール「サイクロン」に新しいアルゴリズムを採用した「サイクロンExpress」が登場しました。今回のバージョンアップの最大のポイントは処理速度の高速化。それでは，そのアルゴリズムを中心に，このソフトの実力を探ってみることにしましょう。

レイトレーシングファン待望のサイクロンExpress(バージョン2.0)がリリースされた。これまでのサイクロン（以下バージョン1)は，モデラー機能は強力だったものの，レンダラのほうはごく普通のレイトレーサであった。特に，速度の点では並の速さでしかなかった。

しかし，このバージョン2.0では大胆なアルゴリズムの見直しを行って，ソフトウェアだけでかなりの高速化を実現している。しかも嬉しいことに，このアルゴリズムでは物体数が増えるほど，計算時間はお得になるのである。そんな魔法のようなアルゴリズムがあるのかとお思いの方に，この魔法のタネをちょっと紹介しておこう。

高速化のタネ明かし

このアルゴリズムの偉大さを理解するためには，一度くらい自分でレイトレーシングのプログラムを書くか，そうでなくても具体的な手順くらいは知っておいたほうがいい。しかし，「使えればそれで結構，速ければそれで結構」という方は，ここは読み飛ばしても支障はないと思う。

ご存じのように，レイトレーシングでは，視点からピクセルごとに発生させたレイ(この場合は視線の意味に考えたほうがわかりやすい）と物体の交点を調べる。もしある物体とレイが交わるなら，その物体は見える（表示される）可能性がある。しかし2つ以上の物体とレイが交わる場合には，そのうちどれが本当に見えるのか（表示されるのか）を決めなくてはならない。その可視・不可視を決めるための尺度として視点と交点の距離を求めておくのである（距離は交点を計算するうちに自然に出てくる)。すると，交点までの距離が小さい，つまり視点からいちばん近い物体が実際に見える(表示される)だろうということは想像がつくであろう。だから，その物体の色を計算してピクセルに書き込み，めでたく1ピクセルの処理は終わる。これがレイトレーシングの本質である。

レイトレーシングは，陰面除去アルゴリズムの中でも力ずくのやり方であるがシーンが正確に描けることは最大のメリットである。しかしよくいわれているように，やっぱり遅いのである。それはなぜだろうか。そう，答えは先の文のなかにある，というのでは不親切だな。ではレイトレーシングのなかに巣くっている遅さの原因とやらを探ってみよう。

たとえば，100個くらいの物体が画面一杯にちりばめてあるシーンを考えてほしい。さて，視点からレイを「ピクセルごとに」発生させる。コンピュータは，100個の物体「すべて」について交点計算をする。ここで2次方程式を解くわけだが，それは決して「軽くはない」処理である。

「　」を付けた部分がキーワードである。もう少し砕いて説明しよう。X 68000の場合，ピクセルは全画面で512×512＝262144個もある。つまりレイはその本数だけ発生する。ではその26万本のレイのうち，あるひとつの物体と交差するレイは何本か？　26万本のうちの5割？　4割？　いやいや，多くて数%，なかには1%を切るものだってザラにある。だって画面上の面積を考えてみてご覧なさい。

それでは，その交差しなかったレイはいったいどうなるのか？　もう想像つくだろう。プログラムは，レイが物体と交差していようがいまいが，物体ごとに用意された2次方程式を空しく解き続けるのである。100個の物体があるシーンなら，1本のレイに対して2次方程式が100個用意される。トータルでは，コンピュータは実に2千万あまりの2次方程式を解かされることになってしまう。しかも，そのうち90%以上はまるっきり無駄な計算なのである。

このことに，どこか不合理なものを感じないだろうか。レイトレーシングが遅いのは，交差判定に数10%の処理時間を喰われていること，そしてそのうちのほとんどが無意味な計算に費やされていること，このためなのである。

あさっての方向を向いているレイは最初から計算しない，たったこれだけのことができれば，数10～数100倍の高速化ができそうなことはもう明らかであろう。で，現在のトレンドともいえる，バージョン2.0のアルゴリズムの登場となるのである。一般には「空間分割法」と呼ばれている。その意味は？　読んで字のごとく，空間を物体の存在する部分と存在しない部分に分けるやり方のことである。

わかりやすい例を出そう。球体は，それより少し大きい立方体で囲める。すると当たり前のことだが，この立方体の外には，球体は決してはみ出すことはない。要するに，この立方体と交差しないレイは球と絶対に交差しない。逆にいえば，球とレイが交差するかどうかを調べるには，まず立方体と交差するかどうかで大雑把に調べておき（これは2次方程式を解くよりもはるかに軽い処理ですむ)，交差する場合だけ改めて2次方程式を解いて正式な交点を求める。多少いい加減な説明だが，こうするだけで，時間のかかる2次方程式の計算回数を大幅に減らせるのである。

この考え方を，もっと一般的な物体空間にまで推し進めたのが空間分割法だと思えばいいだろう。サイクロンExpressでは，空間分割法である「Voxel(ボクセル)分割法」という方式を採用している。

Expressの特徴

さて，バージョン2.0になって，あちらこちらに改良したところが見受けられる。まさにユーザーの声を取り入れた結果であろう。モデラーから順に見ていこう。

これはバージョン1.2からだが，ワイヤーフレーム表示が大幅にスピードアップされている。かなり処理ルーチンを書き直したとのこと。特に数値演算プロセッサを付けた場合は，快適といってもいいレベル。もっと速いに越したことはないが（リアルタ

イム！)。

スピードだけでなく，細かいところにも手を入れてあるのが，使ってみればすぐにわかるだろう。こういった気配りは，カタログデータには決して出てこない。しかし，モデラーの価値を決めるときは，操作性が結構大切なファクターだったりする。

そうそうスピードアップで思い出したが，サイクロンのシステムには，FLOAT？.Xをさらに速くするパッケージが入っている。詳しいことはサイクロンを購入して使ってもらえばわかるだろう(ドキュメントファイルもあるし)。同じ処理なら速いほうが快適なのは世の習いなのだ。

続いてレンダラにいってみよう。バージョン1ではファイルを食わせるといきなりレイトレースを始めたが，バージョン2では，ボクセルのデータを生成するという前処理が入る。モデラーから渡されたデータを見て，空間をどう分割すれば効率よくレイトレースできるかを，前もって計算する過程がはさまる。このため，レイトレースの前に少しだけ待たされる。いってみれば，レイトレース本体の処理を速くするために，前処理にほんの少しシワ寄せしているのである。しかし，レイトレースそのものがとても速くなるのだから，前処理で少しくらい待たされてもまったく気にならないと思う。

それでは気になる「どのくらい速くなるか」である。まず物体数が少ないときは，ボクセル分割の効果があまり出てこない。ボクセルの判定が速いといっても，時間はかかるのだから，プリミティブが少ないうちはそのオーバーヘッドが無視できない。そう，だいたいプリミティブの数が20くらいまでは，バージョン1と同じように計算時間は長くなっていく。

しかし，20あたりを境目として，状況は一変する。なんとそこから先は，プリミティブを増やしていっても，ほとんど計算時間が変わらないのである。もちろん，ボクセル分割の効果である。最初に数が増えるほどお得といったのは，このことなのだ。広告にある最大900倍というのは，決して絵空事ではあるまい。実際，透明体など，1回でピクセルの色が決定できないものはやや時間がかかるが，散乱反射体だけで作られたデータなどはファイルからイメージデータを読み込んでいるのではないかと思うくらい速い。

サイクロンのモデラーは，マクロをばしばしコピー，移動，回転していけるのが売り。その機能が本当においしいのは，このバージョン2.0とコンビを組んだときではなかろうか。マクロやプリミティブをいくら増やしても計算時間が変わらないなら，まさにこのモデラーの本領発揮である。逆に，もはやX68000のレイトレツールは，2，3個の球をちょこんと置いて楽しむレベルをとっくに超えてしまっているともいえる。レイトレのくせに，多いほどいいというのは痛快ではないか。

計算はただでさえ速いが，数値演算プロセッサを付けると，さらに3倍くらいにはなるようだ。なかなかおいしい。

スピードだけではない。ぜひともほしかった機能もバージョン2.0から追加された。たとえばアンチエリアシング。物体の縁にガタガタの線が見えると，なんとなく画像の品質が落ちたように感じたものだ。これで，曲面がきれいに曲面に見えるようになるであろう。

それからとても嬉しいのが，計算の中断と再開ができるようになったこと。バージョン1では，何10時間という計算の間中，X68000は占有されっぱなし。途中でいったん止めて，ワープロやゲームなどをしたくなって，中断したとしよう。その時点で計算は中断ではなく「中止」になってしまい，続きはできなかった。だから中断と再開はぜひともほしかった機能のひとつだった。これさえあれば鬼に金棒，どんな複雑なシーンも恐れずに作れる。ちなみにタイムスタンプの管理はしっかりしているので，重複する処理は起こらない。同じことを何度もやらないようにうまく作ってある。おかげで操作環境は合格点。

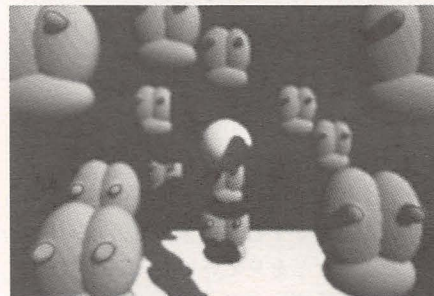
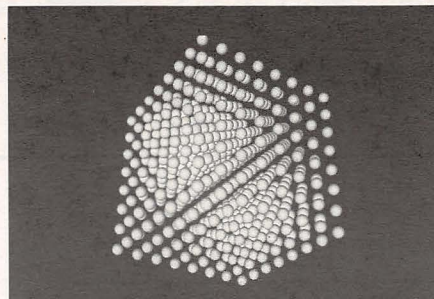
それから，テクスチャ（表面の模様）/バンプ（表面の凹凸）/アトリビュート（反射率，透過率など）のマッピングが可能になった。締め切り間際になって入ってきたサンプル版なので，本格的に使ってみることはできなかったが，マニュアルをパラパラめくってみたところでは，結構器用なこともできそうな感じだ。

豊富なユーティリティ

付属の支援ユーティリティ群のなかにもおいしいものがいくつかあった。

1）デジャギング

アンチエリアシングなしで計算した画像を，アンチエリアシング処理をしたようにいじる。むしろボカシに近いので，アンチエリアシング付きで計算した画像には，画質ではかなわない。

2）圧縮

サイクロンExpress　X68000用 78,000円
アンス・コンサルタンツ　☎092(522)6347

画像ファイルはかなりでっかい。全画面（512×512ドット）だと2HDのフロッピーにも1枚か2枚分しか入らない。これではうまくない。この圧縮プログラムは結構がんばっていて，最高でもとのファイルの1/3ぐらいまでは圧縮する。

3）ディザリング

X68000といえども最大65536色，特にレイトレーシングのような画面ではマッハバンド（色の境界が強調されること）が気になることがある。1600万色のフレームバッファでもあれば問題は解決だが，高い機械だし，誰にでもできるわけではない。そこでこのディザ処理をかければ，マッハバンドが目立たなくなる(意外に効果がある)。ただしこの処理をかけた画像ファイルは圧縮の効率がよくない。

4）マップ作成ツール

マッピングが充実してきたので，Z'sSTAFFライクな操作性のマップエディタも付いている。

5）バージョン1から2へのモデリングデータのコンバータ

レンダラの充実にともなって，モデラーも少々仕様が変更されている。データ構造にも多少の変更があった。そこで，これまでのバージョン1で作ったマクロなどの資産を無駄にしないためのツールだ。

*　　*

以上，駆け足だったが，今回のバージョンアップはなかなか魅力的である。このサイクロンは「レイトレーシングは本当に遅いのか？」という疑問を世に問うことになるソフトなのかもしれない。

大流行中TETRISの快感

TETRISは最強

TETRIS(テトリス)というゲームが流行しています。とりつかれてしまった人が大勢いて，ひまさえあるとやっているようです。

このゲームは，僕の近所だけ，あるいは日本だけで流行っているのではなくて，どうもアメリカを中心として世界中で大流行しているようですね。アメリカのネットワークの記事などを見ていても，TETRISに関する話題はずいぶんとあります。

今ではファミコンや国内のパソコンなどにも移植され，ファンも一段と増えていますが，MacintoshやUNIX版などは，それ以前から出回っていました。Xシリーズ用もかなりのファンを獲得しているようですね。

世界の多くの人たちを魅了しているこのゲームは，見方を変えれば，いま何かと問題となっているコンピュータウイルスそのものであるとみなすことができましょう。実はこれは作者自身がいっていることなのです。要するに，計算機の中に入っていって，CPUやメモリなどの大事な計算機資源を浪費するだけでなく，人の中に入り込んでどっぷり侵してしまうということを意味しているのです。もしかして，あなたもギクッとしたひとりではないでしょうか？ TETRIS菌に侵されて，大切な時間や知能を浪費してはいないでしょうね？

見た目にはきわめて地味なこのゲームは，宣伝攻勢で無理矢理流行らされたものとはわけが違います。相当手ごわいウイルスといえましょう。すでに，「TETRIS現象」とでもいうべきブームを巻き起こした，このゲームの背景を少し冷静に考えてみることにしますか。一般週刊誌にまでTETRISの話題が載っているくらいですからね。なんですって？ TETRISで10万点を取るコツを知りたいんですか（僕にも教えてください，69000点取るコツなら知っていますけど）？

ソ連からアメリカへ，そして日米で裁判

このゲームの興味深いことのひとつに原作者はソ連の人であるということがあります。確かに，Mac版などでもBGMとしてロシア民謡がいい味を出しています。最初に考え出した人の名前はアレクシィ・パジトノフといいまして，作ったときは30歳，計算機関係の研究所（会社？）に勤めていたようです。プログラムを作ったのは別の若者で，18歳のモスクワ大学の学生だそうです。パジトノフ氏が勤めている研究所とアメリカのソフト会社が共同で，まずIBM-PC用に売り出したとのことです（ちょっとあいまいなところもありますが)。

その後の普及ぶりはすでに触れたとおりですが，著作権をめぐる裁判沙汰も，また話題になっています。とにかく，やってみないとこのゲームの魅力はわからないでしょう。ただし，熱くなりやすい人たちと一緒に競い合うのはやめたほうがいいでしょう，感染してしまいます。ワクチンを射ってあって免疫のある人はいいんですけれど。

パソコンだけでなく，UNIXマシン上にもありますし，ファミコン上にもあります。もちろん，Xシリーズにも出ています。本誌の5月号の「読者が選ぶゲームベスト10」でも堂々の一位を獲得していますね。X68000版に続いてX1版も出て刺激になったのでしょう。しかし，どのマシンに移植されても，すばらしいバックミュージックがあるかないかの差はあっても，実に見た目はシンプルだということには変わりがありません。

ゲームの内容について少し述べてみましょう。要するに，画面の上のほうから落ちてくるブロックを，下に落ちる前に左右に移動したり回転させながら，並べていくというものです。そして，横一段に空白なく埋まるとその一段が消されるが，そうでないと次第にブロックが積み重なってしまい，てっぺんに届いたらゲームオーバーというわけです。

ところで，TETRISのバリエーションとしてNLITHというものが巷には出回っているようです。両者は，落ちてくるブロックの種類とか，面クリアなどの点で異なりますが，いちばん大きな違いはNLITHにはへびがいる，ということでしょうか。

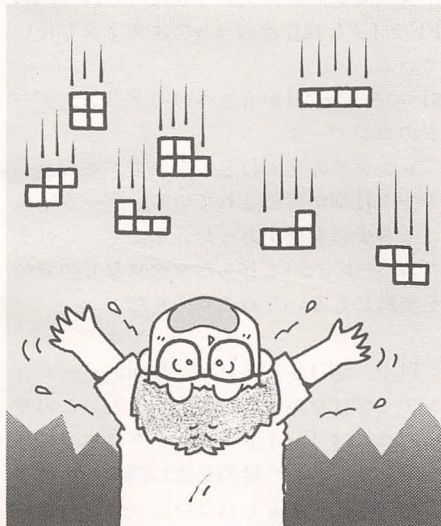
僕としては，へびの挙動をつかむことが大きな試練であり，また面白いところでもあると思ってしまうのですが，TETRISファンにとってはへびの存在は邪道そのものかもしれませんね。

また，NLITHには祈るためのキーがある，ということも大きな違いといえるでしょう。このキーを使うと，スコアは減るものの落ちてくるブロックの形を無理矢理変えることができます。実は，NLITHではこのキーの使いかたが大きなポイントであるようです。

ところで，任天堂のゲームボーイ用のTETRISは2人対戦モードがあるそうです(派手にCMもやっていますよね)。実際にプレイしてみたことはないのですが，話によると自分が一段消すと相手のほうに一段加わるそうです。もし一方が4段いっぺんに消すと，もう一方には4段分が下からにょきっと加わるのです。これではあっという間に終わってしまうような気もしますがどうでしょうか。

エントロピー減少快感法則

なぜTETRISがこんなに面白いのか？ といく前に，突如として，エントロピーについての話をしたいと思います。エントロピーとは何か？ 厳密な話は抜きにして，ひとことでいえば，「無秩序性のめやす」，もっと簡単には，「乱雑さ」とでもいえるでしょう。エントロピーの持つ法則としまして，「エントロピー増大法則(熱力学第2法則)」というものがあります。とにかく，自然の変化というものは，この乱雑さがどうしても減らない方向であるということが示されているのです。

ここで，僕が名づけるところの「エントロピー減少快感法則」なるものが登場する

のです。ごく簡単な例で説明しましょう。僕たちは，ごちゃごちゃになっている部屋を整頓すると，自然に「あー，すっきりした」という気持ちになります。もちろん人によっては，きれいになりすぎて落ち着かないということもありますが，いずれにせよ，ごちゃごちゃと乱雑だったものが，すっきりきれいになった瞬間は，多かれ少なかれ（普遍的とも思えるような）快感を味わえるといえましょう。

ここで，この部屋がごちゃごちゃと散らかっている状態，そしてきちんと整理されている状態との差違が表していることそのものが，エントロピーときわめて近い概念だと考えるわけです。

こうしたことは，この例のような物理的なことに限らず，精神的な面についてもいえるかもしれません。あることの本質がわかれば，ほかのこともそれで説明ができる（抽象的すぎますか？）ということは非常に気持ちのいいことです。まあ学問の進歩なんて，突き詰めればここらへんの意識の奥底にあるのかもしれません。もちろん，この場合エントロピーといっても物理的なものではなく，情報のエントロピーということになります。

2つの実例を出してみたわけですが，おわかりいただけましたでしょうか？　要するに，ほっておくとこの世の中，なんでもエントロピーは増大してしまうのだが，無理にそれを逆方向，つまりエントロピーが減る方向にもっていくと，人はなにゆえか，心地よく感じてしまうのだという仮説を少しリキを入れて主張してみたわけです。そして，この法則を，僕は「エントロピー減少快感法則」と呼ぶのであります。

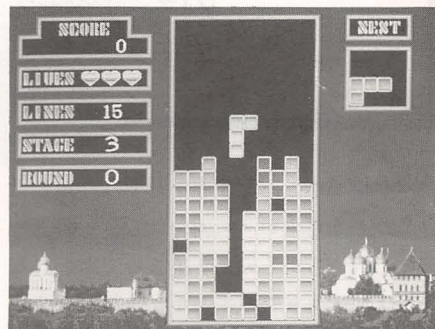
閑話休題

ある大学入試の模試で，僕の連載をヒントにしたら模範解答になったという北海道の方の手紙が，本誌2月号168ページに載っていましたが，このたびその模範解答集を見ることができました。人工知能は進化の次の段階であるというトピックスなど，確かに参考にしていることがわかります。そこのところの話は，僕としてもリキを入れていたところなので，感慨深いものがあります。その読者も，希望している大学に入ってから，僕が取り上げている「人工知能の未来像と人間」に関して，研究を深めてくれるといいですね。

TETRISとエントロピー

TETRIS（あるいはNLITH）を一生懸命やっていることは，エントロピーを減少させるという作業にほかならないと僕は思うのです。降ってくるブロックをがたがたに落とさずに，なるべく平らに落としていく。そしてブロックを消していって，高く重なるのを防ぐ。ヘビという外乱要因が出現すると，なるべくでこぼこにならないように迅速につぶす。

そして，一番下のほうにペタッと並ぶようにブロックの重なる段を消したときの心地よさ，この気持ち良さの原因は，人が本能的，潜在的に持っている，エントロピー減少快感法則を如実に表しているのだといいたいのです（厳密には位置エネルギーと組み合わせて説明する必要があるとは思いますけど）。

もし，ここで僕がまるででっちあげたように述べたてていることをもう少し説得力あるものにするには，まだまだやることがあります。たとえば，TETRISのプレイヤープログラムを作ってみるのです。エントロピーという概念を具体的に画面でのブロックの状態に対する評価関数として求めてみます。画面の上に次のブロックが現れるとプレイヤープログラムはすぐにそのブロックを落としたときの評価関数として計算

X68000版TETRIS

して，プレイヤーの操作を決めるのです。そうして，たぶんそれほど弱くはないプレイヤープログラムが作れると思います。

次に証明しなければならないのは，エントロピー減少快感法則です。もしかしたら，それに似たことはとっくに心理学などの分野でいわれていることかもしれません。それならば，あらためて証明する必要はなくなります。

「エントロピー減少快感法則」，名前は少しものものしい気がしますが，ごく基本的なものであり，生物の本能にも近いものだと思います。そしてTETRISはそれをきわめて忠実にシミュレートしたゲームということができると思います。だからこそ，この単純なゲームは，人の本能に直接感染しながら，ソ連，アメリカ，日本と世界中に流行してきたといえるのでしょう。

あちらこちらで，TETRISだっ！

その1　早稲田大学のNLITH

先日，電子情報通信学会の計算機アーキテクチャ研究会にいったとき，早稲田大学の人が発表していました。そして，提案している方式の名前がなんとNLITHなので，大受けしてしまいました。ちなみに，N Lane Instruction THread computerを略したということです。がんばってますね。

その2　パナソニックのTETRIS

松下通信が人材募集のために凝ったことをしています。「活字メディアでは伝えられない」として，Macのハイパーカードで，アドベンチャーゲームもどきの研究所紹介をしているのです。部門ごとに分かれて紹介するところがありますが，それぞれにコーヒーブレイクという遊びの部分があります。その部分は全部見てしまいましたが，やはりTETRISもどきも登場しました。もっとも，あまり面白いというものでもなかったのですが。

その3　ハイスコアとガッツポーズ

何人かがハイスコアを争い出すと，それはもう熾烈なことになります。ハイスコアを出したときの感激は本人以外には味わえないほどのものです。それまでのハイスコア保持者を懸命に探し，いない場合にはガッツポーズをあたり構わずしまくりながら，その画面をそのままとっておくのです。

その4　NLITHとゲンかつぎ

NLITHのゲームの開始のときに，"今日は満月だ" とかいうメッセージが画面に出るのですが，中にはこのメッセージが出たときのほうがいい点が出やすいと思う人がいまして，MS-DOSのDATEコマンドで，このメッセージが出るような日付けに設定してから，ゲームを開始する人がいます。内蔵の時計を読んでメッセージを出しているのかどうか，それからそのメッセージが出たほうが点が出やすいのか疑問ですが，非常にメンタル(?)なゲームであることを物語っているには違いありません。

第38回

猫とコンピュータ

πの星空

Takazawa Kyoko
高沢 恭子

パソコン使ってこんなことしてみました，と仲間が集うホビーマイコンショウ。キョウコさんのレポートを読むと参加した皆さんがほんとにコンピュータを好きなんだな，と感じます。さて，今回はどんな作品が？

パソコンでこんなことをしてみたんですけど，どうですか？　面白いでしょう。

この気持ちがそのまま集まって開かれるのがホビーショウです。ホビーだから楽しくやります。パソコンにこういうことをさせてみました，あるいはパソコンをこんな色に塗ってみましたなんていう展示もあるのです。

5月下旬のキラキラの日曜日，秋葉原のラジオ会館8階大ホールでは，早朝から重たい機材やそれぞれの愛機が運び込まれ，ケーブルを交差させながら，ワイワイと第5回ホビーマイコンショウのためのセッティングが始まります。

マシン集合

今回の出品は，共催の3つのグループ(きまぐれコンピュータクラブ，FORSIGHT，FBI-NET）から約30点ほど，その中にはFBIのナカムラ隊長企画の「トークセッション」も含まれている。

ショウ開催の大きな利点は，メンバーの交流をさらに深めることはもちろん，見学に訪れるたくさんの人たちの中から毎回新しい知己を得られること，そうした中から貴重なヒントや閃きにますます弾みをつけるところにあるようだ。

今回も500名以上の入場者と，約70名のメンバーの参加を得，その中には名古屋，金沢，群馬から出品作といっしょに上京してきた仲間もいて，ショウの活気は一段と高められた。

メンバーのより深い交流を図るという意味では，準備期間中のたくさんの打ち合わせ（おもに通信ネット上で）の段階から，すでにホビーショウは始まっている。

出品内容や展示の方法，会場内の制約や条件，スペースの分割や足りない端末などの調達，搬入の方法，駐車場の確保，演出上のアイデア，そういったあれこれの質疑応答が交わされながら当日を迎える。

マシンの運搬もそれぞれの作戦で行われるが，夜明けと共にメンバーの家を車で巡ってくれる人(CLOVIS君)などもいて，チームワークの基盤も固い。

広島の中学校の先生イマオカさんは，作品のみの参加なので，例年どおり事前に宅配便でわが家に届けられたものを，夫が当日自分の作品と一緒に会場入りさせる。

会議用の長いデスクをつなげて，壁に沿ってぐるりと展示台ができあがると，いよいよ機材が運び込まれる。トークセッションのためのマイクミキサーを運ぶのは，ミキシング担当のFOL，President氏。「多回線BBSシミュレーション」のために，ホストコンピュータをはじめ拡張装置や8台もの端末を運び入れるナカムラ隊長，SSKさん，カワムラさん以下のFBI軍団。特別参加のヨコマチさんは，迷彩色の塗装をほどこした「手作り・サイケデリックコンピュータ」をエッサエッサと運ぶ。過日，SONYが募った「21世紀のコンピュータ：夢を現実に」という懸賞論文に入賞した折の賞状も展示するそうだ。

コヤマさんの「ジャンケンゲーム」，コバヤシ先生ご夫妻と共に早起きしてやってきた人気者「踊る人形」。そして，X1turboを輝かしいショッキングピンクに塗り上げたわがCHAGAMA会長は，ついでにアスキー社から入手した直径70cmのお化けディスケットをかついで現れて，「どうやって壁に取り付けましょうかぁ」と尋ねている。

また，ホシノさん（MSX），aquerian氏(X1turbo;この日はCHAGAMA氏のマシンを使った）の通信ホストシステム，イシイさんのアーケードゲームなど。こうして各種のマシンが一堂に集合して，みんな喜色満面で小走りに飛び回り，あちらをつなげこちらを結び，実際に機械の息を通わせるまでの期待の時間が，ほんとは一番のショウタイムなのかもしれない。

CGの花

パソコンがものも言わずに（まあ，実際にはキーインする音や，いくらかやかましいプリントアウトの音をあげながら）疲れも見せずに働いてみせることに，私たちはなんだか慣れっこになってしまった。

モニタの画面の中の文字列や記号をながめても，選択と実行でどんなに便利なことができても，そこに人間のワザがあるのを忘れて，パソコンがパソコンらしいことをやっているのだと，なんとなく思ってしまうようになった。

パソコンで描いた幾何学図形を見ても，それがキカイの仕業だとしか感じなかったり，π15万桁の数字が5000ずつのブロックできれいにプリントアウトされていたって，キカイが計算して印刷した数字の行列だ，と思うだけになったりする。そんなとき，それはまったく5000の数字を単体にした格子柄なのだ。

ショウの作品の配置には，毎回だいたいのレギュラー位置があって，入り口を入ってすぐの右手の壁際には，いつもきまぐれコンピュータクラブのワカマツさんの作品が展示される。石川県の高校で数学の先生をされているワカマツさんのコンピュータグラフィックは，主に幾何学的な図形を扱ったもので，毎年その端正な気品がショウをいっそう引き立ててくれる。たくさんのマシンとコードがはいまわり，壁には作品についてのコメントや掲示物ばかりという中で，このコーナーはとても貴重だ。

これらのコンピュータグラフィックは，ワカマツさんのパソコンのテーマのひとつでもあり，毎回彼の出品作の中に含まれて

いる。第４回の作品はFM16βを使って，その名も「グラフィックファンタジー」。抽象画を思わせる折り紙曲線のバリエーションは，第４回ホビーショウの全員の作品をまとめた『パソコンテクノコレクション』（新星出版社刊）の表紙を飾った。

そして第５回では，「グラフィックファンタジー PART2」。与えられた数値により規則的な変化をしながら繰り返しや回帰を続ける点の連続線が，ふしぎに美しい形を描き出す。それはワカマツさんご自身がいうように，「人の手によるものとはまったく異なった独特の美しさ」だ。確かに機械の仕事であるとはいえる。

でも今回の作品「XYプロッタによる習作」を改めて鑑賞したとき，やはり機械を超えたワカマツさんの創意に，はっきりと出会った思いがした。直線，曲線，折れ線が，数式による運動の軌跡で，クリスタルのような透明な花になったり，自由な星群や小宇宙になったりしている。プロッタの描線の太さが一定なのと，色数も多すぎないのがよけい清涼感を与える。これを機械に描かせるには，たくさんの経験もさることながら，創意に基づいてイメージを設計しなければならないのだ。ワカマツさんはパソコンを使って，繊細で壮大な無限を描いているようにみえた。

そう思いながら，ワカマツさんのもうひとつの作品「π157,780桁」を算出したものをながめてみた。計算だけで21時間10分45秒，プリントアウトにさらに30分以上かかったというものが，CGの10倍もの壁面を使って貼られている。行けども行けども数字の海だ。

無限をたどる人

ワカマツさんは1979年のRAMの10月増刊号に，「πを求める」懸賞問題の応募作として，TK-80によってπ100桁までの算出プログラムを発表している。当時，実行時間が30秒という記録もある。それから1982年の第２回ホビーショウには50,000桁までを，翌年はＭＺ-80Ｂで71,616桁までの計算を発表した。

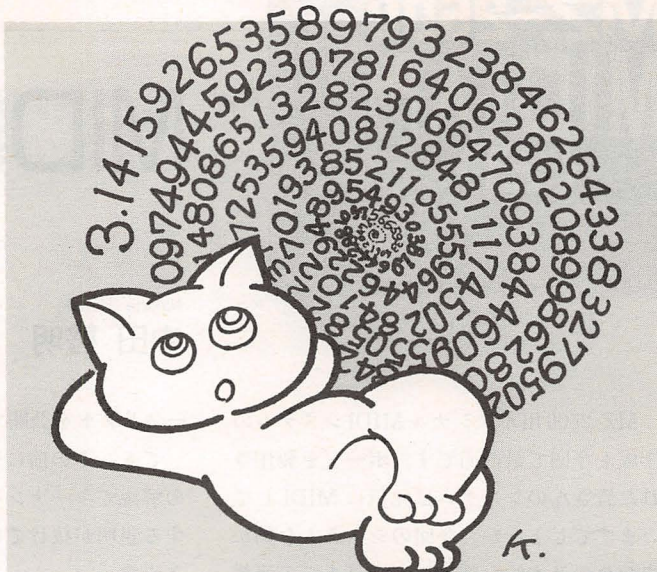
それがパソコンを手に入れて以来のひとつの目標だったとワカマツさんにいわせる「円周率」とは，どうも文盲ならぬ数盲の私が考えているようなものとはチト宇宙が違うらしい。

数字の海を見ながらそう気がついて，遅まきながら円周率について少しばかり調べてみた。読者の皆さんにはすっかりおなじみのことだと思うが，アルキメデスの小数点以下２桁から始まって，今年のスーパーコンピュータでの４億桁まで，汗と涙の奮闘の足跡があった。ほんとうに一生をこの計算に捧げた人というのもいたようだ。Oh! ＭＺ1986年10月号，Oh!X 1988年８月号でもすでに取り上げられているが，πの計算はコンピュータの能力を測る目安にもなりテーマとしてもひとつの分野をなすらしい。

それにしても，円周率というのは円周の直径に対する比率であるとたやすく口にしていたことの，なんと無知だったことか。そもそも有限である円周に無限の計算を挑む気の遠くなるようなエネルギーのすごさ。ガウスの公式，マーチンの公式，ストーマーの公式なんていうのをながめたら，それだけでなんだか荘厳な気分になってくる。

まあ，数学を専門とされているワカマツさんが数字や公式を扱うことはあたりまえとしても，ここではパソコンにそれを計算させるという，さらに１歩進んだことをしているわけだ。10年以上も一貫してそれに取り組み，より優れた方法を追求しながらπの無限をたどる人。そんなワカマツさんのプログラミングの技術を支えてきた最も大きなものは，やはり創意によるインスピレーションなのだろう。

駄菓子屋さんとベルばらと

前回のショウから呼び物のひとつになったのが，天婦羅★三杯酢さんの「駄菓子屋」さんだ。もちろんほんとうの駄菓子を売るお店で，今回は労作のプログラムでパソコンの「クジ占い」も楽しめる。とかく「マシン」の感触が主流になりがちな会場内で，「踊る人形」と並ぶほほえましい出品になっている。

そして新企画の「FBIトーキングセッション'89」。ナカムラシスオペ企画，IB-PATA氏司会，テクニカルスタッフはアベさんとFOL氏。パネラーとしては，プログラマのカワムラさん，メンバーのとりまとめ役を務めるMINE(ミネ)氏，そして私の計３名。ブームマイク，スタンドマイクも準備されて本格的なパネルディスカッションのセッティングだ。

テーマは日頃のパソコンへの接し方や考え方など自分とパソコンの関わりについて話してみようというもの。会場内にハンドマイクを運んでアシスタントを務めてくれたのは，天婦羅氏の妹さんサチコさんと，その親友ハツネさん。ふたりはFBIのマスコットガールで，この日の扮装は「ベルばら」のカップルを思わせる軍服とウエディングドレス。器用なサチコさんが既製服をアレンジしたものだ。ふたりとも長身でなかなか可憐だが，ボーイッシュなヘアスタイルのハツネさんをほんとの男のコと思っていた人もいたらしい。

楽しい演出とPATA氏の巧みな運びでトークの広場は参加者全員が協力したパフォーマンスを作り上げた。

大盛況のショウだった。

テクニカルライターのぶんさんの「無制限VTR予約システム」には取材の方たちも注目していた。千代田・常盤マイコンクラブからもBBSの総帥ともいわれるヨコタさんが特別参加してくださった。「OS/2試運転」と題して出品され，ほんとに試運転が始まるのに午後３時すぎまでかかったのがまさにショウだった。

MZ-2500 MIDI入門 (3)

MIDI用鍵盤表示システム

MZ-2500用MIDIシステムの掲載も今回でひと段落。MIDI 用の鍵盤表示プログラムと全ソースリストを掲載します。この連載で基本システムは揃いました。今度はあなたがこれらを使ったデータを作成する番です。MIDIでの投稿などもお待ちしています。

Nakata Hiroaki
中田 啓明

MZ-2500用オリジナルMIDIシステムの作成も今回で最終回です。ボードを製作された皆さんのシステムは元気にMIDIしていますでしょうか。今回のシステムを動かすための基本的な部分は，先月までの連載で扱ったものだけで十分だといえます。今回は最後にMIDIシーケンサのアクセサリとサンプルデータおよび全プログラムのソースリストを公開することにしましょう。

リスト1 KENBAN

```
C000 C3 41 C0 C3 6A C0 C3 22 : 96
C008 C1 01 01 07 06 05 04 03 : DC
C010 02 01 07 06 05 04 03 02 : 1E
C018 01 07 06 F5 3E 02 F3 D3 : 09
C020 B4 3A B5 00 32 40 C0 3E : 13
C028 02 D3 B4 3E 38 D3 B5 FB : 82
C030 F1 C9 F5 3E 02 F3 D3 B4 : 69
C038 3A 40 C0 D3 B5 FB F1 C9 : 77
C040 20 F5 E5 D5 C5 21 00 C8 : 7D
C048 11 01 C8 01 FF 07 36 00 : 17
C050 ED B0 3E 01 32 09 C0 3E : 15
C058 39 32 2C C0 CD 1B C0 CD : CC
C060 00 C1 CD 32 C0 C1 D1 E1 : F3
C068 F1 C9 F5 E5 D5 C5 D9 E5 : EC
C070 D5 C5 D9 3A 09 C0 3D 32 : E5
C078 09 C0 20 42 3A 0A C0 32 : 61
----------------------------------
SUM: 8E 47 BE 3E 6F 68 53 AD 210B

C080 09 C0 3E 38 32 2C C0 CD : 2A
C088 1B C0 D9 21 18 F7 11 18 : 0D
C090 C8 D9 21 90 41 11 0B C0 : 6F
C098 06 10 C5 D5 E5 1A E6 07 : 9C
C0A0 F6 10 32 F5 C0 CD C8 C0 : 42
C0A8 D9 01 80 00 09 EB 09 EB : 42
C0B0 D9 E1 01 50 00 09 D1 13 : F8
C0B8 C1 10 DF CD 32 C0 D9 C1 : 09
C0C0 D1 E1 D9 C1 D1 E1 F1 C9 : B8
C0C8 06 50 D9 E5 D5 D9 D9 1A : B5
C0D0 FE 05 38 02 3D 12 F3 4E : CD
C0D8 79 E6 7F 77 FB EB 28 09 : 6C
C0E0 79 E6 80 28 06 36 0F 18 : 6A
C0E8 02 36 00 7E EB 23 13 D9 : B0
C0F0 87 77 CB DC 36 16 CB 9C : 58
C0F8 23 10 D3 D9 D1 E1 D9 C9 : 33
----------------------------------
SUM: CE 2A 16 4A 41 D6 E8 BB 54F4

C100 21 00 48 06 10 C5 05 28 : 71
C108 05 36 00 23 10 FB C1 C5 : EF
C110 3E 10 90 28 06 47 36 7E : 07
C118 23 10 FB 36 00 23 C1 10 : 58
C120 E4 C9 F5 E5 DF 0E 28 0C : A8
C128 2A 0A FF 7C B5 28 05 CD : 5E
C130 6A C0 18 F0 CD 6A C0 E1 : 0A
C138 F1 C9 00 00 00 00 00 00 : BA
C140 00 00 00 00 00 00 00 00 : 00
C148 00 00 00 00 00 00 00 00 : 00
C150 00 00 00 00 00 00 00 00 : 00
C158 00 00 00 00 00 00 00 00 : 00
C160 00 00 00 00 00 00 00 00 : 00
C168 00 00 00 00 00 00 00 00 : 00
C170 00 00 00 00 00 00 00 00 : 00
C178 00 00 00 00 00 00 00 00 : 00
----------------------------------
SUM: F0 B2 DF D8 87 CA AA 35 1052
```

でも，その前に先月やったワークエリアの解説でキーナンバーパッチテーブルに関する説明が抜けていたので少々補足しておきます。

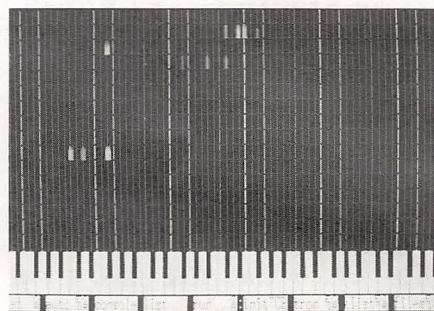
キーナンバーパッチテーブル

MIDI楽器ではパーカッションなどの音程を持たない楽器音をある特定の音程に割り当て，まとめてリズムキットなどの楽器名としてあることが多いのですが，どの音がどの音程に割り当てられているかというのは楽器によってマチマチになっています。たとえば，ハンドクラップではM1ではO4B，D-10ではO2D＋に割り当てられているといったぐあいです（初期状態）。

データにあわせてリズムキットを再編成したりデータをすべて書き換えていくのも大変ですので，シーケンサレベルでこの違いを吸収できるようにしたものが，キーナンバーパッチテーブルです。

パッチテーブルは［SMODE 2］のとき，ED00H～ED7FH，［SMODE 3］のとき，ED80H～EDFFHとなっており，演奏時にこのテーブルの（先頭アドレス）＋（キーナンバー）の部分に格納された内容をMIDIに出力するように指定することができます。

どういう風に使うかというと，要するに他機種用のデータを使うときにはこの部分をまるごとその楽器用に差し換えてやるわけです。

そのほか，ドライバでのテンポ設定範囲が32～255となっているのにマクロでは30～255の範囲になっているなど，不思議に思った方もいるかもしれません。これは実用上ドライバでの設定可能範囲の制限はゆるく，20くらいまで使っても演奏は可能です。しかし，一応の目安としてきりのよい

16チャンネルにフル対応

32としておきましたが，マクロ側ではX1用MIDI MMLと仕様をあわせたため，くい違いが出てきてしまったのです。

また，設定範囲で1～65535となっているパラメータについては，あまり大きな値を入れると誤差が出てしまうので注意が必要です（32000以上くらい）。

どうしても音が出ないという人は，ポートの確認をしてください。今回のドライバはMIDIボードのBポート側にしか出力していません。

KENBANシステム

ではMZ-2500 MIDIシステム用のアクセサリプログラムを発表しましょう。曲データの演奏にあわせて画面上の鍵盤が動作するという，よくあるタイプのアプリケーションです。

従来のMZ-2500拡張MML用やX1用，X68000のOPMA用に発表されているものとは違い，アタックの時点でのインジケータの振幅が最大になるように表示していますので，連続する音符でもはっきりと識別できます。わかりやすくいうと，「音が鳴り始めるとピョンとインジケータの棒が伸び，時間経過とともにゆっくり短くなっていく」わけです。ただし，これは実際のエンベロープ変化とは直接関係ありませんので注意

してください（とても同期できるわけもないのですが)。

これはマシン語（KENBAN）とBASIC（KENBAN.X）で書かれています。

このうちBASIC部分は画面の初期化とマシン語の初期化ルーチンを呼び出すことをやっています。マシン語部分は以下の３つのルーチンから成り立っており，これらのルーチンはCALL文により呼び出されます。

1)　KENBANの初期化

実行アドレス　　　C000H

KENBANのワークエリアの初期化とPCGの設定を行います。

2)　表示（１回のみ）

実行アドレス　　　C003H

現在 MIDI に出力されている各キーの状態を１回だけ表示します。

3)　表示

実行アドレス　　　C006H

C003HをSHIFT＋BREAKされるか演奏が終了するまで呼び出し続けます。ただし，必ず演奏中に呼び出すようにしてください。さもないと場合によっては無限ループに入ります。万一，無限ループに陥ったときはSHIFT＋BREAKしてください。

入力方法

まず，マシン語部分を入力します。

CLEAR ＆HC000

を実行してマシン語エリアを確保してください。このとき，PLAYERなどですでに，

CLEAR ＆HA000

が実行されているならば，マシン語エリアの再確保は不要です。そしてモニタのMコマンドやMACINTO-Cなどのマシン語入力ツールなどからKENBANを正確に入力してください。入力が終わったらC000HからC139Hを実行番地C000Hでセーブしてください。

BASICの初期化サブルーチンはBASIC-M25からそのまま入力しセーブしておきましょう。このとき，MZ-2500はBTXファイルでもマージできますので特にBSDファイルでセーブする必要はありません。

使い方

まず，BASIC部分は演奏プログラムにマージして演奏直前にサブルーチンとして呼び出します。初期化を実行して，演奏開始直後にC006Hをコールすれば演奏中画面に演奏の様子が表示されます。

具体的な方法については，今月のサンプルプログラム，メヌエットとラルゴを見てもらえればわかると思います。鍵盤表示したいときは注釈部分を外し，初期化ルーチンをマージして実行です。

表示はMIDIの１～16chのキーナンバー24～103の80鍵分のみで，それ以外は表示されません。上から順にMIDIの１～16chを示します。

それから，このプログラムは演奏第一に考えられているため，割り込みに同期して表示しているわけではないので，テンポの速い曲や細かな演奏をしているときは，表示が追いつかないことがありますので，ご了承ください。

＊　　＊　　＊

さて，これで３回に渡って発表したMZ-2500用MIDIシステムも打ち止めです。あとはユーザーの皆さんのがんばり次第。今後はLIVE in '89でこのシステム用のプログラムが発表されるのを楽しみにしています。

リスト2　初期化サブルーチン

```
30000 *INIT_KENBAN
30019 BLOAD "KENBAN"
30020 INIT "CRT2:640,200,16":INIT "CRT:80,25,1,0":CLS 3
30030 OY=168:WY=191:BY=182:SY=40
30040 XX=-1
30050 LINE (0,OY)-(639,WY),15,BF
30060 FOR I=0 TO 50
30070 J=I MOD 7
30080 IF J=0 OR J=2 OR J=3 OR J=6 THEN X2=XX+12 ELSE X2=XX+16
30090 LINE (XX,OY)-(X2,WY),0,B:XX=X2
30100 NEXT
30110 XX=7
30120 FOR I=0 TO 35
30130  J=I MOD 5
30140  LINE (XX+1,OY)-(XX+7,BY),0,BF
30150  IF J=1 OR J=4 THEN XX=XX+24 ELSE XX=XX+16
30160 NEXT
30170
30180 FOR I=0 TO 79
30190  IF I MOD 12=0 THEN
30200   C=10
30210  ELSE IF I MOD 12=4 THEN
30220   C=14
30230  ELSE IF I MOD 12=7 THEN
30240   C=13
30250  ELSE
30260   C=9
30270  END IF
30280  LINE (I*8+3,SY)-(I*8+3,OY),C
30290 NEXT
30300 FOR I=SY+7 TO SY+134 STEP 8
30310  LINE (0,I)-(639,I),4
30320 NEXT
30330 CALL $C000
30340 RETURN
```

リスト3　メヌエット

```
10 '*********************************************************************
20 '*                                                                   *
30 '*        メヌエット                                                  *
40 '*                          For MZ-2500+PLAYER.X+MIDI IF+MT-32       *
50 '*                                                                   *
60 '*********************************************************************
70 CLEAR $A000
80 DEF INT A-Z:DIM A$(1,35)
90 DEF USR0=&HD000        'マクロ言語分析
100 DEF USR1=&HD003       '第１トラック入力開始
110 DEF USR2=&HD006       '次のトラック入力開始
120 DEF USR3=&HD009       'すべてのトラックの内容消去＆演奏強制終了
130 DEF USR4=&HD00C       '曲の最初から演奏開始
140 DEF USR5=&HD00F       '停止した演奏の再開
150 DEF USR6=&HD012       '演奏の停止
160 POKE $FF82,1,2,3,4,5,6,7,8
170 FOR I=$B300 TO $B37F:POKE I,0:NEXT
180 FOR I=$B380 TO $B3FF:POKE I,127:NEXT
190 FOR I=0 TO 127:POKE $B328+I/1.5,I:NEXT
200 REM gusub *INIT_KENBAN
210 IF PEEK($FF30)=$55 THEN A=USR3(0)
220 A=USR1(0)
230 A=USR("Y[$F0,$41,$10,$16,$12,$10,$00,$04,16,16,0,0,0,0,0,0,0,$4C,$F7]")
240 RESTORE *MUSIC:CC=0
250 READ A$
260 REPEAT
270  FOR I=0 TO 1
280   A$(I,CC)=A$:READ A$
290  NEXT
300  CC=CC+1
310 UNTIL A$="*"
320
330 FOR I=0 TO 1
340  IF I>0 THEN A=USR2(0)
350  FOR J=0 TO CC-1
360   A=USR(A$(I,J))
370  NEXT
380 NEXT
390 '
400 A=USR4(-1)
410 REM call&hc006
420 END
430 *MUSIC
440 'メヌエット
450 DATA "[MONO]@P30{@12T120@V100L4","[MONO]@P90{@13T120@V100L4"
460 DATA "O6C<F8G8A8A#8","O4F2G"
470 DATA ">C<FF","A2."
480 DATA ">D<A#32A32A#16>C8D8E8","A#2."
490 DATA "F<FF","A2."
500 DATA "A#32A32A#8.>C8<A#8A8G8","G2."
510 DATA "AA#8A8G8F8","F2."
520 DATA "EF8G8A8F8",">C<AF"
530 DATA "A16G8.&G2",">CC8<A#8A8G8"
540 DATA ">C<F8G8A8A#8","A2G"
550 DATA ">C<FF","FAF"
560 DATA ">D<A#32A32A#16>C8D8E8","A#2."
```

```
570 DATA "F<FF","AA#8A8G8F8"
580 DATA "A#32A32A#8.>C8<A#8A8G8","G2>E"
590 DATA "AA#8A8G8F8","F2<A"
600 DATA "GA8G8F8E8","A#>CC"
610 DATA "F2.}","F2<F}"
620 '
630 DATA "{>AF8G8A8F8","{>F2."
640 DATA "GC8D8E8C8","E2."
650 DATA "FD8E8F8C8","DFD"
660 DATA "<BA8B8G","G2<G"
670 DATA "G8A8B8>C8D8E8",">G2."
680 DATA "FED","A>C<B"
```

```
690 DATA "E<GB",">C<EG"
700 DATA ">C2.",">C<C<A#"
710 DATA "C<F8E8F","A>C<A"
720 DATA ">D<F8E8F","A#>D<A#"
730 DATA ">C<A#A","AGF"
740 DATA "G8F8E8F8G",">C2R4"
750 DATA "C8D8E8F8G8A8","C2E"
760 DATA "A#A32A#32A8.G","DFE"
770 DATA "A8>C8<FE","F<A>C"
780 DATA "F2.}","FC<F}"
790 DATA *
800 END
```

リスト4 ラルゴ

```
10 '******************************************************************
20 '*                                                                *
30 '*       Largo(緩叙唱)                                             *
40 '*                           For MZ-2500+PLAYER.X+MIDI IF+MT-32  *
50 '*                                                                *
60 '******************************************************************
70 CLEAR $A000
80 DEF INT A-Z
90 DEF USR0-&HD000        'MACRO SET
100 DEF USR1-&HD003       'OPEN MACRO & FIRST TRACK INPUT
110 DEF USR2-&HD006       'NEXT TRACK INPUT
120 DEF USR3-&HD009       'END MACRO & PLAYER
130 DEF USR4-&HD00C       'START
140 DEF USR5-&HD00F       'CONTINUE
150 DEF USR6-&HD012       'STOP
160 POKE $FF82,2,3,1
170 '
180 REM gosub *INIT_KENBAN
190 IF PEEK($FF30)-$55 THEN A-USR3(0)
200 A-USR1(0)
210 FOR I-1 TO 3
220 IF I>1 THEN A-USR2(0):IF A<>I THEN STOP
230 READ A$:IF A$<>"//" THEN A-USR(A$):GOTO 230
240 NEXT
250 A-USR4($7)
260 CALL &HC006
270 END
280 'PIANO RIGHT HAND
290 DATA "T65@P64@K-3@V80@4905$E2.L4<G:G:G:"
300 DATA ">L4&E<A:>D<A:>C8.:8.<B16:16"
310 DATA "B2.:2."
320 DATA ">C:D<A:GL12>E:F:G:"
330 DATA "L2.FD:"
340 DATA "@<0,95,2.L4>$C2<FD:FD:FD>&C8.:8.<B16:16"
350 DATA "@>0,80,2.B2.E<B:>E<B:>E<B:"
360 DATA "@<0,95,2.>A2D<B:>D<B:>A8.D<B:8.>G16:16"
370 DATA "@>0,80,2.G2.E<B:>E<B:>E<B:"
380 DATA ">>E<GE:>E<AE:>D<BE:"
390 DATA ">D<AE:>C8.<E:8.B16:16BD:"
400 DATA "B8E:8+-A8:8AE:AE:"
410 DATA "@<0,90,4B2.$F:@>0,80,2-A&F:GE:"
420 DATA "A8F8C:8G8E8:8F4.E<F:A>D:8E8:8"
430 '
440 DATA "{E2<G2:2R4:4"
450 DATA ">Q7G<B:>G<B:>G<B:Q8"
460 DATA ">D<B:>D<B:>D<A:"
470 DATA ">E2.<G:G:G:"
480 DATA ">E<B:>D<B:>C8.<E:8.B16:16"
490 DATA "B2.G:G:G:"
500 DATA ">C:D<A:G>L12E:F:G:"
510 DATA "L4.FD:L8EC:L4FD:"
520 DATA "@<0,95,2.>C<E:>C<E:F<B:"
530 DATA "@>0,80,2.>E<B:>E<B:>E<B:"
540 DATA ">ECR:EC<A2:>D<B:"
550 DATA ">E<BG:>E<BG:>>$E<G:"
560 DATA ">&E<A2:>D:C<D:"
570 DATA "B2E2:2BE:"
580 DATA "AE:AF:GE:"
590 DATA "F2C:D:E:"
600 DATA ">D<A:>D<A:>C<G:"
610 DATA "+-B2F2:2>C<E:"
620 DATA ">C2<A:D:+-BF:|"
630 DATA "@<0,95,2.>C<E:AF:AE:"
640 DATA "AD:GE:GE:"
650 DATA "F2D2:2>$C<$D:"
660 DATA ">&C<&D:BE:AD:"
670 DATA "@<0,105,2.GE:BGE:>D<AD:"
680 DATA ">E<BE:>E<BE:>D<BE:"
690 DATA ">D<AE:>C<E:BD:"
700 DATA "BE:+-AE:AE:"
710 DATA "B2.D2:2E:"
720 DATA "@<0,115,2.L8-AF:GE:L4FD:>D<BF:"
730 DATA "W4>E2.<E2.B:@>0,80,2A8:8G8:8W4A:"
740 DATA "FC:GE<B:>FD<A:"
750 DATA "@<0,125,2.>E<G:>BGE:>D<AD:"
760 DATA ">E<BE:>E<BE:>D<BE:"
770 DATA "@>0,95,1>D<AE:>C8.<E:8.B16:16BD:"
780 DATA "B8.E:8.+-A16:16@<0,100,2AE:AE:"
790 DATA "@>0,80,2.B4.F4.<B:>D:8>C8<-A8:8BGE:"
800 DATA "A8F8C:8G8E8:8F4.E<F:>D<A:8>E8:8}"
810 DATA "E2<G2:2R4:4"
820 DATA //
830 'PIANO (LEFT HAND)
840 DATA "@K-3@P64@V80@5503Q7L4BE:BE:BE:"
850 DATA "BF:BF:>_D<F:"
860 DATA ">E<G:>E<G:>E<G:"
870 DATA "Q8>E<A:BF:BE:"
880 DATA "Q7B<B:>B<B:>B<B:"
890 DATA "Q8@<0,95,2.>A:A:A:"
900 DATA "@>0,80,2.G:G:G:"
910 DATA "@<0,95,2.F:F:F:"
920 DATA "@>0,80,2.E:E:E:"
930 DATA ">E<E:>F<F:>G<G:"
940 DATA ">A<A:A:B:"
950 DATA ">C:C:C:"
960 DATA "@<0,90,4D2:@>0,80,2:E:"
970 DATA "<A:B:<B:"
980 '
990 DATA "{>E2:2R4:4"
1000 DATA "Q704L4E:E:E:"
1010 DATA "Q8<B:B:B:"
1020 DATA ">C:C:C:"
1030 DATA "<G:G:A:"
1040 DATA ">E<E:>E<E:>E<E:"
1050 DATA ">E<A:F:E:"
1060 DATA "B2.:2."
1070 DATA "@<0,95,2.A:A:A:"
1080 DATA "@>0,80,2.G:G:G:"
1090 DATA "A:F:B:"
1100 DATA "E:E:>E<E:"
1110 DATA ">F<F:>F<F:>F<F:"
1120 DATA ">G<G:>G<G:>G<G:"
1130 DATA ">C:D:E:"
1140 DATA "<A:>A:G:"
1150 DATA "F:F:E:"
1160 DATA "D:G:A:"
1170 DATA "F:G:<G:"
1180 DATA "@<0,95,2.A:F:F:"
1190 DATA "-B:>E:<E:"
1200 DATA "B:B:A:"
1210 DATA ">A:G:F:"
1220 DATA "@<0,105,2.E:E<E:>F<F:"
1230 DATA ">G<G:G:G:"
1240 DATA "A:A:B:"
1250 DATA ">C:C:C:"
1260 DATA "<B:>-A<-A:>G<G:"
1270 DATA "@<0,115,2.>A:B:<B:"
1280 DATA ">C2.:4@>0,80,2:2"
1290 DATA "<A:B:<B:"
1300 DATA "@<0,125,2.>E:>E<E:>F<F:"
1310 DATA ">G<G:>E<G:>E<G:"
1320 DATA "@>0,95,1>E<A:>A<A:>F<B:"
1330 DATA ">EC:@<0,100,2EC:EC:"
1340 DATA "@>0,80,2.D:<B:>E:"
1350 DATA "<A:B:<B:}"
1360 DATA ">E2:2R4:4"
1370 DATA //
1380 'SONG
1390 DATA "@P64@K-3@12@V80L2.@N100"
1400 DATA "R:R:R:R:R:R:R:"
1410 DATA "R:R:R:R:R:R:R:"
1420 DATA "{L4R:@<0,95,!28805$B2:2"
1430 DATA "$&B2.:2."
1440 DATA "&B:G:@>0,80,1F8.:8.E16:16"
1450 DATA "E2.:2."
1460 DATA "L2.R:"
1470 DATA "R:"
1480 DATA "L4C:D:L12E:F:G:"
1490 DATA "L4F4.:4.E8:8F:"
1500 DATA "@<0,95,2.>C:C:D:"
1510 DATA "E4.:4.@>0,80,4.<B8:8B:"
1520 DATA "R8:8>C8:8<A4.:4.G8:8"
1530 DATA "G2:2$E:"
1540 DATA "&E:D:C8.:8.<B16:16"
1550 DATA "B2.:2."
1560 DATA ">A:A:G:"
1570 DATA "F4.:4.E8:8E:"
1580 DATA "@<0,95,!288>D:D:C:"
1590 DATA "L32<+-B:>C:<B:>C:<B:>C:<B:>C:<B8:8>C8:8L4C:|"
1600 DATA "@V80R8:8<F8:8D4.:4.C8:8"
1610 DATA "C:@<0,95,2.$A2:2"
1620 DATA "&A:G8:8F8:8G:"
1630 DATA "F8.:8.E16:16F:@<0,102,2>$C:"
1640 DATA "&C:<B:@>0,95,2.A8.:8.G16:16"
1650 DATA "G2:2R:"
1660 DATA "@V105>E:E:D:"
1670 DATA "D:C8.:8.<B16:16B:"
1680 DATA "B8:8+-A8:8A4.:4.A8:8"
1690 DATA "B4.:4.>C8:8<B:"
1700 DATA "@<0,115,2.-A8:8G8:8F4.:4.>D8:8"
1710 DATA "E2.:2."
1720 DATA "@V80L12<F:G:A:L4G:F8.:8.E16:16"
1730 DATA "E2.:2."
1740 DATA "L2.R:R:R:R:R:}R:"
1750 DATA //
```


▶ 山田純二君もいつのまにか有名人だなぁ。元同級生の私はそう思う。それにしてもX68000の宣伝がない。さあみんなS.B.S.のもとに集おう。　奥田　健二（19）神奈川県

リスト5 PLAYERソースリスト

```
0000                 1           ;***** MULTI TRACK MUSIC PLAYER *****
E000                 2           ORG 0E000H
0042 P               3   SIOD    EQU 42H
0043 P               4   SIOC    EQU 43H
0044 P               5   CTC0    EQU 44H
0045 P               6   CTC1    EQU 45H
0046 P               7   CTC2    EQU 46H
0047 P               8   CTC3    EQU 47H
E000                 9
0950 P              10   INT1    EQU 0950H
0988 P              11   INT2    EQU 0988H
0019 P              12   DBUFH   EQU 019H
FF30 P              13   OPENF   EQU 0FF30H
0042 P              14   PUSHPOP EQU 0042H
E000                15
E000                16           ;***** INTRUPPT CONTROL SUB *****
E000 CD 42 00       17           CALL PUSHPOP
E003 21 7F E0       18           LD HL,TIMERS             ;CTC SET
E006 11 88 09       19           LD DE,INT2
E009 01 57 00       20           LD BC,TIMERE-TIMERS
E00C ED B0          21           LDIR
E00E 3E E8          22           LD A,0E8H                ;CTC INTRUPPT VECTOR
E010 D3 44          23           OUT (CTC0),A
E012 21 88 09       24           LD HL,INT2               ;CTC INTRUPPT ADRESS
E015 22 EE 02       25           LD (02EEH),HL
E018                26
E018 21 4B E0       27           LD HL,SIOS               ;SIO SET
E01B 11 50 09       28           LD DE,INT1
E01E 01 34 00       29           LD BC,SIOE-SIOS
E021 ED B0          30           LDIR
E023 3E 45          31           LD A,45H                 ;CTC SET FOR SIO CLOCK
E025 D3 44          32           OUT (CTC0),A
E027 3E 03          33           LD A,03H                 ;(6MHz=03H  4MHz=02H)
E029 D3 44          34           OUT (CTC0),A
E02B 21 50 09       35           LD HL,INT1               ;SIO INTRUPPT ADRESS
E02E 22 E0 02       36           LD (02E0H),HL
E031 21 41 E0       37           LD HL,SIOCD
E034 01 43 0A       38           LD BC,0A00H+SIOC
E037 ED B3          39           OTIR
E039 AF             40           XOR A
E03A 32 6D E1       41           LD (AP1),A
E03D 32 30 FF       42           LD (OPENF),A
E040 C9             43           RET
E041                44
E041                45
E041 18 18 02 E0    46   SIOCD   DB 18H,18H,02H,0E0H,04H,44H,03H,0C1H,05H,0E8H
E045 04 44 03 C1
E049 05 E8
E04B                47           ;                  ^SIO INTRUPPT VECTOR
E04B                48
E04B                49
E04B                50           ;***** INTRUPPT SUB (INT1-) *****
E04B F5             51   SIOS    PUSH AF
E04C E5             52           PUSH HL
E04D C5             53           PUSH BC
E04E 3A 81 09       54           LD A,(DCOUNT+INT1-SIOS)
E051 B7             55           OR A
E052 28 13          56           JR Z,DZERO
E054 3D             57           DEC A
E055 32 81 09       58           LD (DCOUNT+INT1-SIOS),A
E058 3A 82 09       59           LD A,(AP2+INT1-SIOS)
E05B 6F             60           LD L,A
E05C 26 19          61           LD H,DBUFH
E05E 3C             62           INC A
E05F 32 82 09       63           LD (AP2+INT1-SIOS),A
E062 7E             64           LD A,(HL)
E063 D3 42          65           OUT (SIOD),A
E065 18 0C          66           JR DZERONE
E067 21 7E 09       67   DZERO   LD HL,SIOR+INT1-SIOS
E06A 01 43 03       68           LD BC,0300H+SIOC
E06D ED B3          69           OTIR
E06F AF             70           XOR A
E070 32 83 09       71           LD (IF+INT1-SIOS),A
E073 C1             72   DZERONE POP BC
E074 E1             73           POP HL
E075 F1             74           POP AF
E076 FB             75           EI
E077 ED 4D          76           RETI
E079                77
E079 01 00 28       78   SIOR    DB 01H,00H,28H
E07C 00             79   DCOUNT  DB 0
E07D 00             80   AP2     DB 0
E07E 00             81   IF      DB 0
E07F                82
E07F                83   SIOE    ;NOP
E07F                84
E07F                85
E07F                86           ;***** INTRUPPT SUB (INT2-) *****
E07F F5             87   TIMERS  PUSH AF
E080 3A DE 09       88           LD A,(WINTF-TIMERS+INT2)
E083 B7             89           OR A
E084 20 43          90           JR NZ,TIEND1
E086 3C             91           INC A
E087 32 DE 09       92           LD (WINTF-TIMERS+INT2),A
E08A E5             93           PUSH HL
E08B D5             94           PUSH DE
E08C C5             95           PUSH BC
E08D DD E5          96           PUSH IX
E08F FD E5          97           PUSH IY
E091 3E 01          98           LD A,1
E093 21 D6 09       99           LD HL,BUFF-TIMERS+INT2
E096 01 B5 07      100           LD BC,07B5H
E099 F3            101           DI
E09A D3 B4         102           OUT (0B4H),A
E09C ED B2         103           INIR
E09E 3E 06         104           LD A,6
E0A0 D3 B4         105           OUT (0B4H),A
E0A2 3A 85 05      106           LD A,(0585H)
E0A5 D3 B5         107           OUT (0B5H),A
E0A7 3A 87 05      108           LD A,(0587H)
E0AA D3 B5         109           OUT (0B5H),A
E0AC FB            110           EI
E0AD CD 99 E1      111           CALL MUSIC
E0B0 3E 01         112           LD A,1
E0B2 21 D6 09      113           LD HL,BUFF-TIMERS+INT2
E0B5 01 B5 07      114           LD BC,07B5H
E0B8 F3            115           DI
E0B9 D3 B4         116           OUT (0B4H),A
E0BB ED B3         117           OTIR
E0BD FB            118           EI
E0BE FD E1         119           POP IY
E0C0 DD E1         120           POP IX
E0C2 C1            121           POP BC
E0C3 D1            122           POP DE
E0C4 E1            123           POP HL
E0C5 AF            124           XOR A
E0C6 32 DE 09      125           LD (WINTF-TIMERS+INT2),A
E0C9 F1            126   TIEND1  POP AF
E0CA FB            127           EI
E0CB ED 4D         128           RETI
E0CD               129
E0CD               130   BUFF    DS 8
E0D5 00            131   WINTF   DB 0
E0D6               132
E0D6               133   TIMERE  ;NOP
E0D6               134
E0D6               135
E0D6               136           ;***** MIDI OUT SUB *****
0981 P             137   DC1     EQU DCOUNT+INT1-SIOS
E0D6               138
E0D6 F5            139   OUT     PUSH AF
E0D7 E5            140           PUSH HL
E0D8 C5            141           PUSH BC
E0D9 47            142           LD B,A
E0DA FE 80         143           CP 80H
E0DC 38 16         144           JR C,OUTGO2
E0DE 3A 6E E1      145           LD A,(EXF)
E0E1 FE 01         146           CP 1
E0E3 28 0F         147           JR Z,OUTGO2
E0E5 B7            148           OR A
E0E6 28 08         149           JR Z,OUTGO1
E0E8 B8            150           CP B
E0E9 20 05         151           JR NZ,OUTGO1
E0EB 78            152           LD A,B
E0EC FE F0         153           CP 0F0H
E0EE 38 2D         154           JR C,OUTPASS
E0F0 78            155   OUTGO1  LD A,B
E0F1 32 6E E1      156           LD (EXF),A
E0F4 3A 83 09      157   OUTGO2  LD A,(IF+INT1-SIOS)
E0F7 B7            158           OR A
E0F8 28 0B         159           JR Z,NOINT
E0FA FB            160           EI
E0FB 21 6C E1      161           LD HL,DC2
E0FE 3A 81 09      162   OWAIT   LD A,(DC1)
E101 86            163           ADD A,(HL)
E102 3C            164           INC A
E103 28 F9         165           JR Z,OWAIT
E105 F3            166   NOINT   DI
E106 3A 6D E1      167           LD A,(AP1)
E109 6F            168           LD L,A
E10A 26 19         169           LD H,DBUFH
E10C 70            170           LD (HL),B
E10D 3C            171           INC A
E10E 32 6D E1      172           LD (AP1),A
E111 3A 6C E1      173           LD A,(DC2)
E114 FE E0         174           CP 0E0H
E116 30 09         175           JR NC,OUTBYTE
E118 FB            176           EI
E119 3C            177           INC A
E11A 32 6C E1      178           LD (DC2),A
E11D C1            179   OUTPASS POP BC
E11E E1            180           POP HL
E11F F1            181           POP AF
E120 C9            182           RET
E121               183
E121 3A 81 09      184   OUTBYTE LD A,(DC1)
E124 3C            185           INC A
E125 32 81 09      186           LD (DC1),A
E128 CD 2D E1      187           CALL OTSTART
E12B 18 F0         188           JR OUTPASS
E12D               189
E12D FB            190   OTSTART EI
E12E 3A 83 09      191           LD A,(IF+INT1-SIOS)
E131 B7            192           OR A
E132 C0            193           RET NZ
E133 3D            194           DEC A
E134 32 83 09      195           LD (IF+INT1-SIOS),A
E137 3E 01         196           LD A,01H
E139 D3 43         197           OUT (SIOC),A
E13B 3E 02         198           LD A,02H
E13D D3 43         199           OUT (SIOC),A
E13F 3A 81 09      200           LD A,(DC1)
E142 3D            201           DEC A
E143 32 81 09      202           LD (DC1),A
E146 3A 82 09      203           LD A,(AP2+INT1-SIOS)
E149 6F            204           LD L,A
E14A 26 19         205           LD H,DBUFH
E14C 3C            206           INC A
E14D 32 82 09      207           LD (AP2+INT1-SIOS),A
E150 7E            208           LD A,(HL)
E151 D3 42         209           OUT (SIOD),A
E153 C9            210           RET
E154               211
E154 F5            212   ALLOUT  PUSH AF
E155 E5            213           PUSH HL
E156 3A 6C E1      214           LD A,(DC2)
E159 B7            215           OR A
E15A 28 0D         216           JR Z,ALLEND
E15C 21 81 09      217           LD HL,DC1
E15F F3            218           DI
E160 86            219           ADD A,(HL)
E161 77            220           LD (HL),A
E162 CD 2D E1      221           CALL OTSTART
E165 AF            222           XOR A
E166 32 6C E1      223           LD (DC2),A
E169 E1            224   ALLEND  POP HL
E16A F1            225           POP AF
E16B C9            226           RET
E16C               227
E16C 00            228   DC2     DB 0
E16D 00            229   AP1     DB 0
E16E 00            230   EXF     DB 0
E16F 00            231   FELF    DB 0
E170               232
E170               233           ;***** MUSIC PLAYER *****
0139 P             234   MUL     EQU 0139H
0153 P             235   DIV     EQU 0153H
EE00 P             236   TRACK   EQU 0EE00H
0090 P             237   TLEN    EQU 090H
F700 P             238   MIDI    EQU TLEN*16+TRACK
FF00 P             239   SYSTEM  EQU MIDI+800H
FFA0 P             240   RTBUFF  EQU 0FFA0H
E170               241
ED00 P             242   PATCH1  EQU 0ED00H                ;PATCH AREA FOR RHYTHM P
ART
ED80 P             243   PATCH2  EQU 0ED80H
E170               244
E170               245           ;***** DIV 24bit (ABC¥DE->ABC...HL) *****
E170 08            246   DIV24   EX AF,AF'
E171 F5            247           PUSH AF
E172 3E 18         248           LD A,24
E174 21 00 00      249           LD HL,0
E177 08            250   DIV24LP EX AF,AF'
E178 CB 11         251           RL C
E17A CB 10         252           RL B
E17C 17            253           RLA
E17D CB 15         254           RL L
E17F CB 14         255           RL H
E181 B7            256           OR A
E182 E5            257           PUSH HL
E183 ED 52         258           SBC HL,DE
E185 3F            259           CCF
E186 30 04         260           JR NC,DIVBAD
E188 33            261           INC SP .
E189 33            262           INC SP
E18A 18 01         263           JR DIVNE
E18C E1            264   DIVBAD  POP HL
E18D 08            265   DIVNE   EX AF,AF'
E18E 3D            266           DEC A
E18F 20 E6         267           JR NZ,DIV24LP
E191 F1            268           POP AF
E192 08            269           EX AF,AF'
E193 CB 11         270           RL C
E195 CB ,10        271           RL B
```

▶５月号の佐川さん，７月号の倉本さん，僕も奈美ちゃんを応援しています。最近は，テレビではあまり見かけなくなりましたが，がんばってシャープのCFに出てもらおうではあーりませんか。

加藤　佳之（21）神奈川県

```
E197 17            272          RLA
E198 C9            273          RET
E199               274
E199               275          ;***** TOTAL CONTROL *****
E199               276
E199               277  MUSIC   ;* ACTIVE SENS.
E199 3A 0D FF      278          LD A,(SYSTEM+0DH)
E19C 3D            279          DEC A
E19D 20 0F         280          JR NZ,ACTNE
E19F 3E FE         281          LD A,0FEH        ;ACTIVE SENS.
E1A1 CD D6 E0      282          CALL OUT
E1A4 CD 54 E1      283          CALL ALLOUT
E1A7 AF            284          XOR A
E1A8 32 6E E1      285          LD (EXF),A
E1AB 3A 0F FF      286          LD A,(SYSTEM+0FH)
E1AE 32 0D FF      287  ACTNE   LD (SYSTEM+0DH),A
E1B1 21 AC FF      288          LD HL,RTBUFF+0CH
E1B4 35            289          DEC (HL)
E1B5 20 07         290          JR NZ,MUSKIP
E1B7 2B            291          DEC HL
E1B8 7E            292          LD A,(HL)
E1B9 23            293          INC HL
E1BA 77            294          LD (HL),A
E1BB 23            295          INC HL
E1BC 36 FF         296          LD (HL),0FFH
E1BE               297
E1BE               298          ;* FELMATA CHECK
E1BE 2A 1C FF      299  MUSKIP  LD HL,(SYSTEM+1CH)
E1C1 7C            300          LD A,H
E1C2 B5            301          OR L
E1C3 32 6F E1      302          LD (FELF),A
E1C6 28 04         303          JR Z,TRCALL
E1C8 2B            304          DEC HL
E1C9 22 1C FF      305          LD (SYSTEM+1CH),HL
E1CC               306
E1CC               307          ;* TRACK CALL
E1CC DD 21 00 EE   308  TRCALL  LD IX,TRACK
E1D0 21 00 FE      309          LD HL,0FE00H
E1D3 22 10 FF      310          LD (SYSTEM+10H),HL
E1D6 06 10         311          LD B,16
E1D8 ED 5B 0A FF   312          LD DE,(SYSTEM+0AH)
E1DC CB 3A         313  TRLOOP  SRL D
E1DE CB 1B         314          RR E
E1E0 30 07         315          JR NC,TRNE1
E1E2 D5            316          PUSH DE
E1E3 C5            317          PUSH BC
E1E4 CD E7 E2      318          CALL TRACKX
E1E7 C1            319          POP BC
E1E8 D1            320          POP DE
E1E9 2A 10 FF      321  TRNE1   LD HL,(SYSTEM+10H)
E1EC CB 04         322          RLC H
E1EE 38 01         323          JR C,TRNEX
E1F0 2C            324          INC L
E1F1 22 10 FF      325  TRNEX   LD (SYSTEM+10H),HL
E1F4 EB            326          EX DE,HL
E1F5 11 90 00      327          LD DE,TLEN
E1F8 DD 19         328          ADD IX,DE
E1FA EB            329          EX DE,HL
E1FB 10 DF         330          DJNZ TRLOOP
E1FD               331
E1FD               332          ;* SET TEMPO
E1FD 3A 0E FF      333          LD A,(SYSTEM+0EH)
E200 B7            334          OR A
E201 3A 00 FF      335          LD A,(SYSTEM)
E204 C4 96 E2      336          CALL NZ,STEMPO
E207 AF            337          XOR A
E208 32 0E FF      338          LD (SYSTEM+0EH),A
E20B               339
E20B               340          ;* RILATIVE TEMPO
E20B FD 21 A0 FF   341          LD IY,RTBUFF
E20F FD 7E 00      342          LD A,(IY+00H)
E212 FE 80         343          CP 80H
E214 38 13         344          JR C,RITNE1
E216 E6 07         345          AND 07H
E218 FD 77 00      346          LD (IY+00H),A
E21B 3A 00 FF      347          LD A,(SYSTEM)
E21E FD 77 0A      348          LD (IY+0AH),A
E221 FD 6E 02      349          LD L,(IY+02H)
E224 FD 66 03      350          LD H,(IY+03H)
E227 18 40         351          JR RITNE3
E229 FD 7E 00      352  RITNE1  LD A,(IY+00H)
E22C E6 01         353          AND 01H
E22E CA 54 E1      354          JP Z,ALLOUT
E231 FD 6E 08      355          LD L,(IY+08H)
E234 FD 66 09      356          LD H,(IY+09H)
E237 2B            357          DEC HL
E238 FD 75 08      358          LD (IY+08H),L
E23B FD 74 09      359          LD (IY+09H),H
E23E 7C            360          LD A,H
E23F B5            361          OR L
E240 20 16         362          JR NZ,RITNE2
E242 FD 7E 00      363          LD A,(IY+00H)
E245 E6 06         364          AND 06H
E247 FD 77 00      365          LD (IY+00H),A
E24A E6 02         366          AND 02H
E24C FD 7E 01      367          LD A,(IY+01H)
E24F 28 02         368          JR Z,RTSK1
E251 ED 44         369          NEG
E253 FD 86 0A      370  RTSK1   ADD A,(IY+0AH)
E256 18 31         371          JR RITNE6
E258 CD 4E EC      372  RITNE2  CALL CURSUB
E25B D2 54 E1      373          JP NC,ALLOUT
E25E FD 7E 0D      374          LD A,(IY+0DH)
E261 B7            375          OR A
E262 CA 54 E1      376          JP Z,ALLOUT
E265 FD 36 0D 00   377          LD (IY+0DH),0
E269 6E            378  RITNE3  LD L,(HL)
E26A 26 00         379          LD H,0
E26C FD 7E 01      380          LD A,(IY+01H)
E26F 5F            381          LD E,A
E270 54            382          LD D,H
E271 CD 39 01      383          CALL MUL
E274 11 FF 00      384          LD DE,255
E277 CD 53 01      385          CALL DIV
E27A FD 7E 0A      386          LD A,(IY+0AH)
E27D 67            387          LD H,A
E27E FD 7E 00      388          LD A,(IY+00H)
E281 E6 02         389          AND 02H
E283 7D            390          LD A,L
E284 28 02         391          JR Z,RITNE5
E286 ED 44         392          NEG
E288 84            393  RITNE5  ADD A,H
E289 21 00 FF      394  RITNE6  LD HL,SYSTEM
E28C BE            395          CP (HL)
E28D CA 54 E1      396          JP Z,ALLOUT
E290 CD 96 E2      397          CALL STEMPO
E293 C3 54 E1      398          JP ALLOUT
E296               399
E296               400          ;***** SET TEMPO SUB *****
E296 32 00 FF      401  STEMPO  LD (SYSTEM),A
E299 6F            402          LD L,A
E29A 26 00         403          LD H,0
E29C E5            404          PUSH HL
E29D 11 07 00      405          LD DE,7
E2A0 CD 53 01      406          CALL DIV
E2A3 7D            407          LD A,L
E2A4 32 0F FF      408          LD (SYSTEM+0FH),A
E2A7 AF            409          XOR A
E2A8 32 0E FF      410          LD (SYSTEM+0EH),A
E2AB 3C            411          INC A
E2AC 32 0D FF      412          LD (SYSTEM+0DH),A
E2AF D1            413          POP DE
E2B0 3E 07         414          LD A,7
E2B2 01 0E 27      415          LD BC,270EH
E2B5 CD 70 E1      416          CALL DIV24
E2B8 58            417          LD E,B
E2B9 16 00         418          LD D,0
E2BB 13            419          INC DE
E2BC D5            420          PUSH DE
E2BD CD 70 E1      421          CALL DIV24
E2C0 D1            422          POP DE
E2C1 3E 15         423          LD A,15H
E2C3 D3 46         424          OUT (CTC2),A
E2C5 7B            425          LD A,E
E2C6 D3 46         426          OUT (CTC2),A
E2C8 3E D5         427          LD A,0D5H
E2CA D3 47         428          OUT (CTC3),A
E2CC 79            429          LD A,C
E2CD D3 47         430          OUT (CTC3),A
E2CF C9            431          RET
E2D0               432
E2D0               433          ;***** TRACK CONTROL *****
E2D0 32 0C FF      434  INITSET LD (SYSTEM+0CH),A
E2D3 67            435          LD H,A
E2D4 C6 20         436          ADD A,20H
E2D6 32 1A FF      437          LD (SYSTEM+1AH),A
E2D9 2E 00         438          LD L,0
E2DB CB 3C         439          SRL H
E2DD CB 1D         440          RR L
E2DF 11 00 F7      441          LD DE,MIDI
E2E2 19            442          ADD HL,DE
E2E3 22 14 FF      443          LD (SYSTEM+14H),HL
E2E6 C9            444          RET
E2E7               445
E2E7 DD 7E 00      446  TRACKX  LD A,(IX)
E2EA E6 0F         447          AND 0FH
E2EC CD D0 E2      448          CALL INITSET
E2EF DD E5         449          PUSH IX
E2F1 E1            450          POP HL
E2F2 22 16 FF      451          LD (SYSTEM+16H),HL
E2F5 11 1B 00      452          LD DE,1BH
E2F8 19            453          ADD HL,DE
E2F9 35            454          DEC (HL)
E2FA 20 0C         455          JR NZ,TRXSK1
E2FC 2B            456          DEC HL
E2FD 7E            457          LD A,(HL)
E2FE 23            458          INC HL
E2FF 77            459          LD (HL),A
E300 23            460          INC HL
E301 23            461          INC HL
E302 23            462          INC HL
E303 CB C6         463          SET 0,(HL)
E305 2B            464          DEC HL
E306 18 02         465          JR TRXSK3
E308 23            466  TRXSK1  INC HL
E309 23            467          INC HL
E30A 35            468  TRXSK3  DEC (HL)
E30B 20 08         469          JR NZ,TRXSK2
E30D 2B            470          DEC HL
E30E 7E            471          LD A,(HL)
E30F 23            472          INC HL
E310 77            473          LD (HL),A
E311 23            474          INC HL
E312 CB CE         475          SET 1,(HL)
E314 2B            476          DEC HL
E315 11 33 00      477  TRXSK2  LD DE,50H-1DH
E318 19            478          ADD HL,DE
E319 22 18 FF      479          LD (SYSTEM+18H),HL
E31C 3A 6F E1      480          LD A,(FELF)
E31F B7            481          OR A
E320 20 23         482          JR NZ,CUSKIP
E322               483
E322               484          ;* KEY OFF TIMER CONTROL
E322 2A 18 FF      485          LD HL,(SYSTEM+18H)
E325 06 10         486          LD B,16
E327 4E            487  KOFTCLP LD C,(HL)
E328 23            488          INC HL
E329 7E            489          LD A,(HL)
E32A B7            490          OR A
E32B 28 06         491          JR Z,KOFTCNE
E32D 3D            492          DEC A
E32E 77            493          LD (HL),A
E32F 79            494          LD A,C
E330 CC C8 E9      495          CALL Z,KEYOFF
E333 23            496  KOFTCNE INC HL
E334 10 F1         497          DJNZ KOFTCLP
E336               498
E336 CD EC E3      499          CALL C_BEND
E339               500
E339               501          ;* WAIT CONTROL
E339 DD 7E 1F      502          LD A,(IX+1FH)
E33C 3D            503          DEC A
E33D DD 77 1F      504          LD (IX+1FH),A
E340 CC 77 E4      505          CALL Z,READ
E343 18 03         506          JR RILVOL
E345               507
E345 CD EC E3      508  CUSKIP  CALL C_BEND
E348               509
E348               510          ;* RILATIVE VOLUME
E348 FD 2A 16 FF   511  RILVOL  LD IY,(SYSTEM+16H)
E34C 01 80 00      512          LD BC,80H
E34F FD 09         513          ADD IY,BC
E351 FD 7E 00      514          LD A,(IY+00H)
E354 FE 80         515          CP 80H
E356 38 13         516          JR C,RVNE1
E358 E6 7F         517          AND 7FH
E35A FD 77 00      518          LD (IY+00H),A
E35D DD 46 04      519          LD B,(IX+04H)
E360 FD 70 0A      520          LD (IY+0AH),B
E363 FD 6E 02      521          LD L,(IY+02H)
E366 FD 66 03      522          LD H,(IY+03H)
E369 18 40         523          JR RVNE3
E36B E6 01         524  RVNE1   AND 1
E36D C8            525          RET Z
E36E FD 6E 08      526          LD L,(IY+08H)
E371 FD 66 09      527          LD H,(IY+09H)
E374 2B            528          DEC HL
E375 FD 75 08      529          LD (IY+08H),L
E378 FD 74 09      530          LD (IY+09H),H
E37B 7C            531          LD A,H
E37C B5            532          OR L
E37D 20 1F         533          JR NZ,RVNE2
E37F 3E 06         534          LD A,06H
E381 FD A6 00      535          AND (IY+00H)
E384 47            536          LD B,A
E385 FD 77 00      537          LD (IY+00H),A
E388 DD 7E 3E      538          LD A,(IX+3EH)
E38B B7            539          OR A
E38C C2 6D E9      540          JP NZ,E
E38F 78            541          LD A,B
E390 E6 02         542          AND 02H
E392 FD 7E 01      543          LD A,(IY+01H)
E395 28 02         544          JR Z,RVSK1
E397 ED 44         545          NEG
E399 FD 86 0A      546  RVSK1   ADD A,(IY+0AH)
E39C 18 34         547          JR RVNEX
E39E CD 4E EC      548  RVNE2   CALL CURSUB
E3A1 D0            549          RET NC
E3A2 DD CB 1E 46   550          BIT 0,(IX+1EH)
E3A6 C8            551          RET Z
E3A7 DD CB 1E 86   552          RES 0,(IX+1EH)
E3AB 6E            553  RVNE3   LD L,(HL)
E3AC 26 00         554          LD H,0
E3AE FD 5E 01      555          LD E,(IY+01H)
```

▶昨年１年間，受験のためほとんど読めなかったが，浪人したのでもう１年間読めない。２年もほとんど読まないと浦島太郎になってしまいそうだ，どうしよう。情報工学科に行きたいのに。

伊砂　隆志（19）京都府

```
E3B1 54             556            LD D,H
E3B2 CD 39 01       557            CALL MUL
E3B5 11 FF 00       558            LD DE,255
E3B8 CD 53 01       559            CALL DIV
E3BB FD 7E 00       560            LD A,(IY+00H)
E3BE E6 02          561            AND 2
E3C0 FD 7E 0A       562            LD A,(IY+0AH)
E3C3 28 06          563            JR Z,RVNE4
E3C5 95             564            SUB L
E3C6 30 0A          565            JR NC,RVNEX
E3C8 AF             566            XOR A
E3C9 18 07          567            JR RVNEX
E3CB 85             568   RVNE4    ADD A,L
E3CC FE 80          569            CP 80H
E3CE 38 02          570            JR C,RVNEX
E3D0 3E 7F          571            LD A,7FH
E3D2 DD BE 04       572   RVNEX    CP (IX+4)
E3D5 C8             573            RET Z
E3D6 F5             574            PUSH AF
E3D7 3A 0C FF       575            LD A,(SYSTEM+0CH)    ;MIDI OUT VOLUME
E3DA F6 B0          576            OR 0B0H
E3DC CD D6 E0       577            CALL OUT
E3DF 3E 07          578            LD A,7
E3E1 CD D6 E0       579            CALL OUT
E3E4 F1             580            POP AF
E3E5 DD 77 04       581            LD (IX+4),A
E3E8 CD D6 E0       582            CALL OUT
E3EB C9             583            RET
E3EC                584
E3EC                585            ;* PITCH BENDER
E3EC FD 2A 16 FF    586   C_BEND   LD IY,(SYSTEM+16H)
E3F0 01 10 00       587            LD BC,10H
E3F3 FD 09          588            ADD IY,BC
E3F5 FD 7E 00       589            LD A,(IY+00H)
E3F8 E6 01          590            AND 1
E3FA C8             591            RET Z
E3FB FD 6E 08       592            LD L,(IY+08H)
E3FE FD 66 09       593            LD H,(IY+09H)
E401 2B             594            DEC HL
E402 FD 75 08       595            LD (IY+08H),L
E405 FD 74 09       596            LD (IY+09H),H
E408 7C             597            LD A,H
E409 B5             598            OR L
E40A 20 09          599            JR NZ,PBNE2
E40C FD 36 00 00    600            LD (IY+00H),0
E410 21 00 40       601            LD HL,4000H
E413 18 3E          602            JR PBNEX4
E415 CD 4E EC       603   PBNE2    CALL CURSUB
E418 D0             604            RET NC
E419 DD CB 1E 4E    605            BIT 1,(IX+1EH)
E41D C8             606            RET Z
E41E DD CB 1E 8E    607            RES 1,(IX+1EH)
E422 7E             608   PBNE3    LD A,(HL)
E423 16 00          609            LD D,0
E425 62             610            LD H,D
E426 FE 80          611            CP 80H
E428 38 03          612            JR C,PBNEX1
E42A ED 44          613            NEG
E42C 04             614            INC B
E42D 5F             615   PBNEX1   LD E,A
E42E FD 7E 01       616            LD A,(IY+01H)
E431 FE 80          617            CP 80H
E433 38 03          618            JR C,PBNEX2
E435 ED 44          619            NEG
E437 04             620            INC B
E438 6F             621   PBNEX2   LD L,A
E439 C5             622            PUSH BC
E43A CD 39 01       623            CALL MUL
E43D F1             624            POP AF
E43E 0F             625            RRCA
E43F 30 07          626            JR NC,PBNEX3
E441 EB             627            EX DE,HL
E442 21 00 00       628            LD HL,0
E445 B7             629            OR A
E446 ED 52          630            SBC HL,DE
E448 11 00 40       631   PBNEX3   LD DE,4000H
E44B 19             632            ADD HL,DE
E44C 7C             633            LD A,H
E44D FE 80          634            CP 80H
E44F DA 53 E4       635            JP C,PBNEX4
E452 2B             636            DEC HL
E453 DD 5E 0E       637   PBNEX4   LD E,(IX+0EH)
E456 DD 56 0F       638            LD D,(IX+0FH)
E459 EB             639            EX DE,HL
E45A CB 3B          640            SRL E
E45C B7             641            OR A
E45D ED 52          642            SBC HL,DE
E45F C8             643            RET Z
E460 DD 73 0E       644            LD (IX+0EH),E
E463 DD 72 0F       645            LD (IX+0FH),D
E466 3A 0C FF       646            LD A,(SYSTEM+0CH)      ;BENDER OUT
E469 F6 E0          647            OR 0E0H
E46B CD D6 E0       648            CALL OUT
E46E 7B             649            LD A,E
E46F CD D6 E0       650            CALL OUT
E472 7A             651            LD A,D
E473 CD D6 E0       652            CALL OUT
E476 C9             653            RET
E477                654
E477                655            ;***** DATA READ *****
E477 3E 01          656   READ     LD A,1
E479 2A 16 FF       657            LD HL,(SYSTEM+16H)
E47C 01 78 00       658            LD BC,0078H
E47F 09             659            ADD HL,BC
E480 01 B5 05       660            LD BC,05B5H
E483 F3             661            DI
E484 D3 B4          662            OUT (0B4H),A
E486 ED B3          663            OTIR
E488 FB             664            EI
E489 DD 6E 38       665            LD L,(IX+38H)
E48C DD 66 39       666            LD H,(IX+39H)
E48F 22 12 FF       667            LD (SYSTEM+12H),HL
E492                668
E492 2A 12 FF       669   LOOP     LD HL,(SYSTEM+12H)
E495 7E             670            LD A,(HL)
E496 23             671            INC HL
E497 FE 80          672            CP 80H
E499 DA 35 EA       673            JP C,KEYON
E49C E5             674            PUSH HL
E49D E6 7F          675            AND 7FH
E49F 47             676            LD B,A
E4A0 87             677            ADD A,A
E4A1 6F             678            LD L,A
E4A2 26 00          679            LD H,0
E4A4 11 AF E4       680            LD DE,JTABLE
E4A7 19             681            ADD HL,DE
E4A8 5E             682            LD E,(HL)
E4A9 23             683            INC HL
E4AA 56             684            LD D,(HL)
E4AB EB             685            EX DE,HL
E4AC 78             686            LD A,B
E4AD D1             687            POP DE
E4AE E9             688            JP (HL)
E4AF                689
E4AF                690   JTABLE
E4AF AF E5 AF E5    691            DW  A1,A2,PP,P,MP,MF,F,FF,STACC,TENU,TIE,RTIE,RE
POUT,E,E,E
E4B3 B5 E5 BD E5
E4B7 C5 E5 CD E5
E4BB D5 E5 DD E5
E4BF E5 E5 E5 E5
E4C3 E5 E5 EC E5
E4C7 F7 E5 6D E9
E4CB 6D E9 6D E9
E4CF 27 E6 30 E6    692            DW  GATE,REPIN,DC,DS,SENYO,TO,CODA,FINE,KAKKO,NE
IRO,VOL,PAN,EXP,MOD,TEM,E
E4D3 3E E6 53 E6
E4D7 A9 E6 C8 E6
E4DB FE E6 08 E7
E4DF 22 E7 71 E7
E4E3 82 E7 A7 E7
E4E7 BD E7 D3 E7
E4EB E9 E7 6D E9
E4EF 1C E8 6D E9    693            DW  FO,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E
E4F3 6D E9 6D E9
E4F7 6D E9 6D E9
E4FB 6D E9 6D E9
E4FF 6D E9 6D E9
E503 6D E9 6D E9
E507 6D E9 6D E9
E50B 6D E9 6D E9
E50F 85 E8 85 E8    694            DW  CRESC,DIM,ACCEL,RIT,BEND,E,E,E,E,E,E,E,E,E,E
,E
E513 C1 E8 C1 E8
E517 51 E8 6D E9
E51B 6D E9 6D E9
E51F 6D E9 6D E9
E523 6D E9 6D E9
E527 6D E9 6D E9
E52B 6D E9 6D E9
E52F 9C EB A5 EB    695            DW  PPSET,PSET,MPSET,MFSET,FSET,FFSET,NSET,A1SET
,A2SET,CHSET,STSET,E,E,E,E,E
E533 AE EB B7 EB
E537 C0 EB C9 EB
E53B D2 EB DB EB
E53F E4 EB ED EB
E543 FB EB 6D E9
E547 6D E9 6D E9
E54B 6D E9 6D E9
E54F 6D E9 6D E9    696            DW  E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E
E553 6D E9 6D E9
E557 6D E9 6D E9
E55B 6D E9 6D E9
E55F 6D E9 6D E9
E563 6D E9 6D E9
E567 6D E9 6D E9
E56B 6D E9 6D E9
E56F EE E8 FA E8    697            DW  WAIT,WAIT256,CHO,RINGI,MIDIOUT,SHIFT,FEL,CHA
NGE,SMODE,?T,?V,?B,E,E,E,REST
E573 09 E9 00 E9
E577 04 EC 21 EC
E57B 28 EC 33 EC
E57F 47 EC FE E7
E583 09 E8 13 E8
E587 6D E9 6D E9
E58B 6D E9 13 E9
E58F 6D E9 6D E9    698            DW  E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,BAR
E593 6D E9 6D E9
E597 6D E9 6D E9
E59B 6D E9 6D E9
E59F 6D E9 6D E9
E5A3 6D E9 6D E9
E5A7 6D E9 6D E9
E5AB 6D E9 60 E9
E5AF                699
E5AF                700            ;***** ACCENT *****
E5AF                701   A1
E5AF 3C             702   A2       INC A
E5B0 DD 77 3B       703            LD (IX+3BH),A
E5B3 18 3B          704            JR UP0
E5B5                705
E5B5                706            ;***** pp *****
E5B5 DD 7E 08       707   PP       LD A,(IX+08H)
E5B8 CD 86 EB       708            CALL VOLSET
E5BB 18 33          709            JR UP0
E5BD                710
E5BD                711            ;***** p *****
E5BD DD 7E 09       712   P        LD A,(IX+09H)
E5C0 CD 86 EB       713            CALL VOLSET
E5C3 18 2B          714            JR UP0
E5C5                715
E5C5                716            ;***** mp *****
E5C5 DD 7E 0A       717   MP       LD A,(IX+0AH)
E5C8 CD 86 EB       718            CALL VOLSET
E5CB 18 23          719            JR UP0
E5CD                720
E5CD                721            ;***** mf *****
E5CD DD 7E 0B       722   MF       LD A,(IX+0BH)
E5D0 CD 86 EB       723            CALL VOLSET
E5D3 18 1B          724            JR UP0
E5D5                725
E5D5                726   F        ;***** f *****
E5D5 DD 7E 0C       727            LD A,(IX+0CH)
E5D8 CD 86 EB       728            CALL VOLSET
E5DB 18 13          729            JR UP0
E5DD                730
E5DD                731   FF       ;***** ff *****
E5DD DD 7E 0D       732            LD A,(IX+0DH)
E5E0 CD 86 EB       733            CALL VOLSET
E5E3 18 0B          734            JR UP0
E5E5                735
E5E5                736            ;***** STACC,TENU,TIE *****
E5E5                737   STACC
E5E5                738   TENU
E5E5 D6 07          739   TIE      SUB 07H
E5E7 DD 77 4E       740            LD (IX+4EH),A
E5EA 18 04          741            JR UP0
E5EC                742
E5EC DD 36 3A 01    743   RTIE     LD (IX+3AH),1
E5F0                744
E5F0 ED 53 12 FF    745   UP0      LD (SYSTEM+12H),DE
E5F4 C3 92 E4       746            JP LOOP
E5F7                747
E5F7                748            ;***** REPEAT OUT *****
E5F7 1A             749   REPOUT   LD A,(DE)
E5F8 FE 98          750            CP 98H            ;KAKKO
E5FA 20 07          751            JR NZ,REONE
E5FC 13             752            INC DE
E5FD 1A             753            LD A,(DE)
E5FE 13             754            INC DE
E5FF 13             755            INC DE
E600 CD 54 E7       756            CALL KAKKOSET
E603 DD 7E 23       757   REONE    LD A,(IX+23H)
E606 B7             758            OR A
E607 DD 6E 21       759            LD L,(IX+21H)
E60A DD 66 22       760            LD H,(IX+22H)
E60D 20 0B          761            JR NZ,REONE2
E60F 7E             762            LD A,(HL)
E610 B7             763            OR A
E611 28 07          764            JR Z,REONE2
E613 ED 53 12 FF    765            LD (SYSTEM+12H),DE
E617 C3 92 E4       766            JP LOOP
E61A 36 01          767   REONE2   LD (HL),1
E61C DD 36 23 00    768            LD (IX+23H),0
E620 23             769            INC HL
E621 22 12 FF       770            LD (SYSTEM+12H),HL
E624 C3 92 E4       771            JP LOOP
E627                772
E627                773
E627                774            ;***** GATE *****
E627 1A             775   GATE     LD A,(DE)
E628 E6 07          776            AND 7
E62A DD 77 20       777            LD (IX+20H),A
E62D C3 F6 E7       778            JP UP1
```

▶テレビのステレオ放送ってコマーシャルのためにあるんですね。やっぱりスポンサーには，勝てないんだな。　巣山　修一（21）長野県

```
E630                    779
E630                    780              ;***** REPEAT IN *****
E630 DD 73 21           781     REPIN    LD (IX+21H),E
E633 DD 72 22           782              LD (IX+22H),D
E636 AF                 783              XOR A
E637 12                 784              LD (DE),A
E638 DD 77 23           785              LD (IX+23H),A
E63B C3 F6 E7           786              JP UP1
E63E                    787
E63E                    788              ;***** D.C. *****
E63E CD 86 E6           789     DC       CALL CCODA
E641 CD 63 E6           790              CALL DFRAG
E644 DA F6 E7           791              JP C,UP1
E647 DD 6E 36           792              LD L,(IX+36H)
E64A DD 66 37           793              LD H,(IX+37H)
E64D 22 12 FF           794              LD (SYSTEM+12H),HL
E650 C3 92 E4           795              JP LOOP
E653                    796
E653                    797              ;***** D.S. *****
E653 CD 86 E6           798     DS       CALL CCODA
E656 CD 63 E6           799              CALL DFRAG
E659 DA F6 E7           800              JP C,UP1
E65C ED 53 12 FF        801              LD (SYSTEM+12H),DE
E660 C3 92 E4           802              JP LOOP
E663                    803
E663 1A                 804     DFRAG    LD A,(DE)
E664 2A 16 FF           805              LD HL,(SYSTEM+16H)
E667 01 2A 00           806              LD BC,002AH
E66A 09                 807              ADD HL,BC
E66B 06 04              808              LD B,4
E66D 0F                 809     DFLOOP   RRCA
E66E F5                 810              PUSH AF
E66F 30 0D              811              JR NC,DFNE1
E671 7E                 812              LD A,(HL)
E672 B7                 813              OR A
E673 20 09              814              JR NZ,DFNE1
E675 36 01              815              LD (HL),1
E677 23                 816              INC HL
E678 5E                 817              LD E,(HL)
E679 23                 818              INC HL
E67A 56                 819              LD D,(HL)
E67B F1                 820              POP AF
E67C B7                 821              OR A
E67D C9                 822              RET
E67E 23                 823     DFNE1    INC HL
E67F 23                 824              INC HL
E680 23                 825              INC HL
E681 F1                 826              POP AF
E682 10 E9              827              DJNZ DFLOOP
E684 37                 828              SCF
E685 C9                 829              RET
E686                    830
E686 D5                 831     CCODA    PUSH DE
E687 13                 832              INC DE
E688 1A                 833              LD A,(DE)
E689 FE 96              834              CP 96H
E68B 20 1A              835              JR NZ,CCODANE
E68D C5                 836              PUSH BC
E68E 13                 837              INC DE
E68F 1A                 838              LD A,(DE)
E690 2A 16 FF           839              LD HL,(SYSTEM+16H)
E693 01 70 00           840              LD BC,0070H
E696 09                 841              ADD HL,BC
E697 06 04              842              LD B,4
E699 13                 843              INC DE
E69A 0F                 844     CCOLOOP  RRCA
E69B 30 05              845              JR NC,CCONE1
E69D 73                 846              LD (HL),E
E69E 23                 847              INC HL
E69F 72                 848              LD (HL),D
E6A0 18 01              849              JR CCONE2
E6A2 23                 850     CCONE1   INC HL
E6A3 23                 851     CCONE2   INC HL
E6A4 10 F4              852              DJNZ CCOLOOP
E6A6 C1                 853              POP BC
E6A7 D1                 854     CCODANE  POP DE
E6A8 C9                 855              RET
E6A9                    856
E6A9                    857              ;***** ·S· *****
E6A9 1A                 858     SENYO    LD A,(DE)
E6AA 2A 16 FF           859              LD HL,(SYSTEM+16H)
E6AD 01 2B 00           860              LD BC,002BH
E6B0 09                 861              ADD HL,BC
E6B1 13                 862              INC DE
E6B2 06 04              863              LD B,4
E6B4 0F                 864     SELOOP   RRCA
E6B5 30 05              865              JR NC,SENE1
E6B7 73                 866              LD (HL),E
E6B8 23                 867              INC HL
E6B9 72                 868              LD (HL),D
E6BA 18 01              869              JR SENE2
E6BC 23                 870     SENE1    INC HL
E6BD 23                 871     SENE2    INC HL
E6BE 23                 872              INC HL
E6BF 10 F3              873              DJNZ SELOOP
E6C1 ED 53 12 FF        874              LD (SYSTEM+12H),DE
E6C5 C3 92 E4           875              JP LOOP
E6C8                    876
E6C8                    877              ;***** to *****
E6C8 1A                 878     TO       LD A,(DE)
E6C9 2A 16 FF           879              LD HL,(SYSTEM+16H)
E6CC 01 2A 00           880              LD BC,002AH
E6CF 09                 881              ADD HL,BC
E6D0 13                 882              INC DE
E6D1 01 00 04           883              LD BC,0400H
E6D4 0F                 884     TOLOOP   RRCA
E6D5 F5                 885              PUSH AF
E6D6 30 18              886              JR NC,TONE1
E6D8 7E                 887              LD A,(HL)
E6D9 FE 01              888              CP 1
E6DB 20 13              889              JR NZ,TONE1
E6DD 3C                 890              INC A
E6DE 77                 891              LD (HL),A
E6DF 2A 16 FF           892              LD HL,(SYSTEM+16H)
E6E2 11 70 00           893              LD DE,0070H
E6E5 19                 894              ADD HL,DE
E6E6 79                 895              LD A,C
E6E7 87                 896              ADD A,A
E6E8 5F                 897              LD E,A
E6E9 19                 898              ADD HL,DE
E6EA 5E                 899              LD E,(HL)
E6EB 23                 900              INC HL
E6EC 56                 901              LD D,(HL)
E6ED F1                 902              POP AF
E6EE 18 07              903              JR TONE2
E6F0 23                 904     TONE1    INC HL
E6F1 23                 905              INC HL
E6F2 23                 906              INC HL
E6F3 0C                 907              INC C
E6F4 F1                 908              POP AF
E6F5 10 DD              909              DJNZ TOLOOP
E6F7 ED 53 12 FF        910     TONE2    LD (SYSTEM+12H),DE
E6FB C3 92 E4           911              JP LOOP
E6FE                    912
E6FE                    913              ;***** CODA *****
E6FE D5                 914     CODA     PUSH DE
E6FF 1B                 915              DEC DE
E700 1B                 916              DEC DE
E701 CD 86 E6           917              CALL CCODA
E704 D1                 918              POP DE
E705 C3 F6 E7           919              JP UP1
E708                    920
```

```
E708                    921              ;***** Fine *****
E708 1A                 922     FINE     LD A,(DE)
E709 2A 16 FF           923              LD HL,(SYSTEM+16H)
E70C 01 2A 00           924              LD BC,002AH
E70F 09                 925              ADD HL,BC
E710 06 04              926              LD B,4
E712 0F                 927     FINELP   RRCA
E713 30 05              928              JR NC,FINENE
E715 7E                 929              LD A,(HL)
E716 B7                 930              OR A
E717 C2 6D E9           931              JP NZ,E          ;END
E71A 23                 932     FINENE   INC HL
E71B 23                 933              INC HL
E71C 23                 934              INC HL
E71D 10 F3              935              DJNZ FINELP
E71F C3 F6 E7           936              JP UP1
E722                    937
E722                    938              ;***** KAKKO *****
E722 1A                 939     KAKKO    LD A,(DE)
E723 13                 940              INC DE
E724 13                 941              INC DE
E725 CD 54 E7           942              CALL KAKKOSET
E728 1B                 943              DEC DE
E729 DD 36 23 01        944              LD (IX+23H),1
E72D DD 6E 21           945              LD L,(IX+21H)
E730 DD 66 22           946              LD H,(IX+22H)
E733 7E                 947              LD A,(HL)
E734 B7                 948              OR A
E735 28 19              949              JR Z,KAKKOX
E737 1A                 950              LD A,(DE)
E738 3C                 951              INC A
E739 12                 952              LD (DE),A
E73A 87                 953              ADD A,A
E73B 2A 16 FF           954              LD HL,(SYSTEM+16H)
E73E 01 22 00           955              LD BC,22H
E741 09                 956              ADD HL,BC
E742 4F                 957              LD C,A
E743 06 00              958              LD B,0
E745 09                 959              ADD HL,BC
E746 5E                 960              LD E,(HL)
E747 23                 961              INC HL
E748 56                 962              LD D,(HL)
E749 ED 53 12 FF        963              LD (SYSTEM+12H),DE
E74D C3 92 E4           964              JP LOOP
E750 12                 965     KAKKOX   LD (DE),A        ;A=0
E751 C3 F6 E7           966              JP UP1
E754                    967
E754                    968              ;***** KAKKO SET *****
E754                    969     KAKKOSET
E754 F5                 970              PUSH AF
E755 E5                 971              PUSH HL
E756 C5                 972              PUSH BC
E757 2A 16 FF           973              LD HL,(SYSTEM+16H)
E75A 01 24 00           974              LD BC,24H
E75D 09                 975              ADD HL,BC
E75E 0F                 976              RRCA
E75F 06 03              977              LD B,3
E761 0F                 978     KSETLP   RRCA
E762 30 05              979              JR NC,KSETNE
E764 73                 980              LD (HL),E
E765 23                 981              INC HL
E766 72                 982              LD (HL),D
E767 18 01              983              JR KSETNE2
E769 23                 984     KSETNE   INC HL
E76A 23                 985     KSETNE2  INC HL
E76B 10 F4              986              DJNZ KSETLP
E76D C1                 987              POP BC
E76E E1                 988              POP HL
E76F F1                 989              POP AF
E770 C9                 990              RET
E771                    991
E771                    992              ;***** NEIRO *****
E771 3A 0C FF           993     NEIRO    LD A,(SYSTEM+0CH)        ;MIDI PROGRAM CHANGE
E774 F6 C0              994              OR 0C0H
E776 CD D6 E0           995              CALL OUT
E779 1A                 996              LD A,(DE)
E77A DD 77 3D           997              LD (IX+3DH),A
E77D CD D6 E0           998              CALL OUT
E780 18 74              999              JR UP1
E782                   1000
E782                   1001              ;***** VOL *****
E782 FD 2A 16 FF       1002     VOL      LD IY,(SYSTEM+16H)
E786 01 80 00          1003              LD BC,80H
E789 FD 09             1004              ADD IY,BC
E78B FD 36 00 00       1005              LD (IY+00H),0
E78F 3A 0C FF          1006              LD A,(SYSTEM+0CH)        ;MIDI VOL OUT
E792 F6 B0             1007              OR 0B0H
E794 CD D6 E0          1008              CALL OUT
E797 3E 07             1009              LD A,07H
E799 CD D6 E0          1010              CALL OUT
E79C 1A                1011              LD A,(DE)
E79D E6 7F             1012              AND 7FH
E79F DD 77 04          1013              LD (IX+04H),A
E7A2 CD D6 E0          1014              CALL OUT
E7A5 18 4F             1015              JR UP1
E7A7                   1016
E7A7                   1017              ;***** PAN *****
E7A7 3A 0C FF          1018     PAN      LD A,(SYSTEM+0CH)        ;MIDI PAN OUT
E7AA F6 B0             1019              OR 0B0H
E7AC CD D6 E0          1020              CALL OUT
E7AF 3E 0A             1021              LD A,0AH
E7B1 CD D6 E0          1022              CALL OUT
E7B4 1A                1023              LD A,(DE)
E7B5 DD 77 05          1024              LD (IX+05H),A
E7B8 CD D6 E0          1025              CALL OUT
E7BB 18 39             1026              JR UP1
E7BD                   1027
E7BD                   1028              ;***** EXP *****
E7BD 3A 0C FF          1029     EXP      LD A,(SYSTEM+0CH)        ;MIDI EXP OUT
E7C0 F6 B0             1030              OR 0B0H
E7C2 CD D6 E0          1031              CALL OUT
E7C5 3E 0B             1032              LD A,0BH
E7C7 CD D6 E0          1033              CALL OUT
E7CA 1A                1034              LD A,(DE)
E7CB DD 77 06          1035              LD (IX+06H),A
E7CE CD D6 E0          1036              CALL OUT
E7D1 18 23             1037              JR UP1
E7D3                   1038
E7D3                   1039              ;***** MODULATION *****
E7D3 3A 0C FF          1040     MOD      LD A,(SYSTEM+0CH)        ;MIDI MOD OUT
E7D6 F6 B0             1041              OR 0B0H
E7D8 CD D6 E0          1042              CALL OUT
E7DB 3E 01             1043              LD A,01H
E7DD CD D6 E0          1044              CALL OUT
E7E0 1A                1045              LD A,(DE)
E7E1 DD 77 07          1046              LD (IX+07H),A
E7E4 CD D6 E0          1047              CALL OUT
E7E7 18 0D             1048              JR UP1
E7E9                   1049
E7E9                   1050              ;***** TEMPO *****
E7E9 AF                1051     TEM      XOR A
E7EA 32 A0 FF          1052              LD (RTBUFF),A
E7ED 1A                1053              LD A,(DE)
E7EE 32 00 FF          1054              LD (SYSTEM),A
E7F1 3E 01             1055              LD A,1
E7F3 32 0E FF          1056              LD (SYSTEM+0EH),A
E7F6                   1057
E7F6 13                1058     UP1      INC DE
E7F7 ED 53 12 FF       1059              LD (SYSTEM+12H),DE
E7FB C3 92 E4          1060              JP LOOP
E7FE                   1061
E7FE 1A                1062     ?T       LD A,(DE)
```

▶うーん，ハードディスクは今日も元気……と思った瞬間，ハードディスクのファンの音が消えて……。そうです，妻が洗濯機のコンセントと間違えてX68000（ノーマルタイプ）のコンセントを抜いてしまったんです。オートリラクトのハードディスクが欲しいよー。

安石　清太郎（28）広島県

```
E7FF 32 AB FF     1063          LD (RTBUFF+0BH),A
E802 3E 01        1064          LD A,1
E804 32 AC FF     1065          LD (RTBUFF+0CH),A
E807 18 ED        1066          JR UP1
E809              1067
E809 1A           1068  ?V      LD A,(DE)
E80A DD 77 1A     1069          LD (IX+1AH),A
E80D DD 36 1B 01  1070          LD (IX+1BH),1
E811 18 E3        1071          JR UP1
E813              1072
E813 1A           1073  ?B      LD A,(DE)
E814 DD 77 1C     1074          LD (IX+1CH),A
E817 DD 77 1D     1075          LD (IX+1DH),A
E81A 18 DA        1076          JR UP1
E81C              1077
E81C              1078          ;***** F.O. *****
E81C DD 6E 21     1079  FO      LD L,(IX+21H)
E81F DD 66 22     1080          LD H,(IX+22H)
E822 7E           1081          LD A,(HL)
E823 36 01        1082          LD (HL),1
E825 23           1083          INC HL
E826 22 12 FF     1084          LD (SYSTEM+12H),HL
E829 B7           1085          OR A
E82A C2 92 E4     1086          JP NZ,LOOP
E82D FD 2A 16 FF  1087          LD IY,(SYSTEM+16H)
E831 01 80 00     1088          LD BC,80H
E834 FD 09        1089          ADD IY,BC
E836 FD 36 00 83  1090          LD (IY+00H),83H
E83A 1A           1091          LD A,(DE)
E83B CD C0 E9     1092          CALL CURVADR
E83E FD 75 02     1093          LD (IY+02H),L
E841 FD 74 03     1094          LD (IY+03H),H
E844 DD 7E 04     1095          LD A,(IX+04H)
E847 CD 8B EC     1096          CALL RTDSET
E84A DD 36 3E 01  1097          LD (IX+3EH),1
E84E C3 92 E4     1098          JP LOOP
E851              1099
E851              1100          ;***** PITCH BENDER *****
E851 FD 2A 16 FF  1101  BEND    LD IY,(SYSTEM+16H)
E855 01 10 00     1102          LD BC,10H
E858 FD 09        1103          ADD IY,BC
E85A FD 7E 00     1104          LD A,(IY+00H)
E85D B7           1105          OR A
E85E 20 08        1106          JR NZ,BEND2
E860 DD 36 0E 00  1107          LD (IX+0EH),00H
E864 DD 36 0F 40  1108          LD (IX+0FH),40H
E868 1A           1109  BEND2   LD A,(DE)                  ;FRAG SET
E869 CD C0 E9     1110          CALL CURVADR
E86C FD 75 02     1111          LD (IY+02H),L
E86F FD 74 03     1112          LD (IY+03H),H
E872 E5           1113          PUSH HL
E873 13           1114          INC DE
E874 FD 36 00 01  1115          LD (IY+00H),1
E878 1A           1116          LD A,(DE)
E879 CD 8B EC     1117          CALL RTDSET
E87C E1           1118          POP HL
E87D D5           1119          PUSH DE
E87E CD 22 E4     1120          CALL PBNE3
E881 D1           1121          POP DE
E882 C3 F6 E7     1122          JP UP1
E885              1123
E885              1124          ;***** cresc & dim *****
E885              1125  CRESC
E885 1A           1126  DIM     LD A,(DE)
E886 CD C0 E9     1127          CALL CURVADR
E889 FD 2A 16 FF  1128          LD IY,(SYSTEM+16H)
E88D 01 80 00     1129          LD BC,80H
E890 FD 09        1130          ADD IY,BC
E892 FD 75 02     1131          LD (IY+02H),L
E895 FD 74 03     1132          LD (IY+03H),H
E898 13           1133          INC DE
E899 DD 7E 04     1134          LD A,(IX+04H)
E89C 47           1135          LD B,A
E89D 1A           1136          LD A,(DE)
E89E FE 80        1137          CP 80H
E8A0 38 0A        1138          JR C,DIMNNE
E8A2 E6 07        1139          AND 07H
E8A4 2A 16 FF     1140          LD HL,(SYSTEM+16H)
E8A7 C6 08        1141          ADD A,8
E8A9 85           1142          ADD A,L
E8AA 6F           1143          LD L,A
E8AB 7E           1144          LD A,(HL)
E8AC 90           1145  DIMNNE  SUB B
E8AD 38 06        1146          JR C,DIM2
E8AF FD 36 00 81  1147          LD (IY+00H),81H
E8B3 18 06        1148          JR DIMNE
E8B5 FD 36 00 83  1149  DIM2    LD (IY+00H),83H
E8B9 ED 44        1150          NEG
E8BB CD 8B EC     1151  DIMNE   CALL RTDSET
E8BE C3 F6 E7     1152          JP UP1
E8C1              1153
E8C1              1154          ;***** acchel & rit(rall) *****
E8C1              1155  ACCEL
E8C1 1A           1156  RIT     LD A,(DE)
E8C2 CD C0 E9     1157          CALL CURVADR
E8C5 FD 21 A0 FF  1158          LD IY,RTBUFF
E8C9 FD 75 02     1159          LD (IY+02H),L
E8CC FD 74 03     1160          LD (IY+03H),H
E8CF 13           1161          INC DE
E8D0 3A 00 FF     1162          LD A,(SYSTEM)
E8D3 47           1163          LD B,A
E8D4 1A           1164          LD A,(DE)
E8D5 21 A0 FF     1165          LD HL,RTBUFF
E8D8 90           1166          SUB B
E8D9 38 04        1167          JR C,RIT2
E8DB 36 81        1168          LD (HL),81H
E8DD 18 04        1169          JR RITNE
E8DF 36 83        1170  RIT2    LD (HL),83H
E8E1 ED 44        1171          NEG
E8E3 CD 8B EC     1172  RITNE   CALL RTDSET
E8E6 13           1173          INC DE
E8E7 ED 53 12 FF  1174          LD (SYSTEM+12H),DE
E8EB C3 92 E4     1175          JP LOOP
E8EE              1176
E8EE              1177          ;***** WAIT *****
E8EE 1A           1178  WAIT    LD A,(DE)
E8EF DD 77 1F     1179          LD (IX+1FH),A
E8F2 13           1180          INC DE
E8F3 DD 73 38     1181  WAITNE1 LD (IX+38H),E
E8F6 DD 72 39     1182          LD (IX+39H),D
E8F9 C9           1183          RET
E8FA              1184
E8FA DD 36 1F 00  1185  WAIT256 LD (IX+1FH),0
E8FE 18 F3        1186          JR WAITNE1
E900              1187
E900              1188          ;***** RINGI *****
E900 1A           1189  RINGI   LD A,(DE)
E901 C6 80        1190          ADD A,80H
E903 DD 77 4D     1191          LD (IX+4DH),A
E906 C3 F6 E7     1192          JP UP1
E909              1193
E909              1194          ;***** CHO *****
E909 1A           1195  CHO     LD A,(DE)
E90A DD 77 3F     1196          LD (IX+3FH),A
E90D CD 16 E9     1197          CALL CHOSET
E910 C3 F6 E7     1198          JP UP1
E913              1199
E913              1200          ;***** REST *****
E913 C3 F6 E7     1201  REST    JP UP1
E916              1202
E916              1203          ;*CHOSET
E916 F5           1204  CHOSET  PUSH AF
E917 E5           1205          PUSH HL
E918 D5           1206          PUSH DE
E919 C5           1207          PUSH BC
E91A 2A 16 FF     1208          LD HL,(SYSTEM+16H)
E91D 11 40 00     1209          LD DE,40H
E920 19           1210          ADD HL,DE
E921 E5           1211          PUSH HL
E922 01 00 0C     1212          LD BC,0C00H
E925 71           1213  CHOCLP  LD (HL),C
E926 23           1214          INC HL
E927 10 FC        1215          DJNZ CHOCLP
E929 D1           1216          POP DE
E92A B7           1217          OR A
E92B 28 20        1218          JR Z,CHOEND
E92D FE 80        1219          CP 80H
E92F 30 07        1220          JR NC,FLAT
E931 21 52 E9     1221          LD HL,SPOINT
E934 0E 01        1222          LD C,1
E936 18 07        1223          JR FLATNE
E938 21 59 E9     1224  FLAT    LD HL,FPOINT
E93B 0E FF        1225          LD C,255
E93D ED 44        1226          NEG
E93F E6 07        1227  FLATNE  AND 7
E941 47           1228          LD B,A
E942 7E           1229  CHOSLP  LD A,(HL)
E943 E5           1230          PUSH HL
E944 6F           1231          LD L,A
E945 26 00        1232          LD H,0
E947 19           1233          ADD HL,DE
E948 71           1234          LD (HL),C
E949 E1           1235          POP HL
E94A 23           1236          INC HL
E94B 10 F5        1237          DJNZ CHOSLP
E94D C1           1238  CHOEND  POP BC
E94E D1           1239          POP DE
E94F E1           1240          POP HL
E950 F1           1241          POP AF
E951 C9           1242          RET
E952              1243
E952 05 00 07 02  1244  SPOINT  DB 5,0,7,2,9,4,11
E956 09 04 0B
E959 0B 04 09 02  1245  FPOINT  DB 11,4,9,2,7,0,5
E95D 07 00 05
E960              1246
E960              1247
E960              1248          ;***** BAR *****
E960 DD 7E 3F     1249  BAR     LD A,(IX+3FH)
E963 CD 16 E9     1250          CALL CHOSET
E966 ED 53 12 FF  1251          LD (SYSTEM+12H),DE
E96A C3 92 E4     1252          JP LOOP
E96D              1253
E96D              1254          ;***** END *****
E96D 3A 10 FF     1255  E       LD A,(SYSTEM+10H)          ;FRAG RESET
E970 4F           1256          LD C,A
E971 06 00        1257          LD B,0
E973 21 0A FF     1258          LD HL,SYSTEM+0AH
E976 09           1259          ADD HL,BC
E977 3A 11 FF     1260          LD A,(SYSTEM+11H)
E97A 47           1261          LD B,A
E97B 7E           1262          LD A,(HL)
E97C A0           1263          AND B
E97D 77           1264          LD (HL),A
E97E 2A 18 FF     1265          LD HL,(SYSTEM+18H)
E981 06 10        1266          LD B,16
E983 23           1267  ENDLOOP INC HL
E984 36 00        1268          LD (HL),0
E986 23           1269          INC HL
E987 10 FA        1270          DJNZ ENDLOOP
E989 06 80        1271          LD B,80H
E98B AF           1272          XOR A
E98C 2A 14 FF     1273          LD HL,(SYSTEM+14H)
E98F 77           1274  ENDLP2  LD (HL),A
E990 23           1275          INC HL
E991 10 FC        1276          DJNZ ENDLP2
E993 3A 0C FF     1277          LD A,(SYSTEM+0CH)          ;ALL NOTE OFF
E996 F6 B0        1278          OR 0B0H
E998 CD D6 E0     1279          CALL OUT
E99B 3E 7B        1280          LD A,7BH
E99D CD D6 E0     1281          CALL OUT
E9A0 AF           1282          XOR A
E9A1 CD D6 E0     1283          CALL OUT
E9A4 3A 0C FF     1284          LD A,(SYSTEM+0CH)          ;BENDER RESET
E9A7 F6 E0        1285          OR 0E0H
E9A9 CD D6 E0     1286          CALL OUT
E9AC AF           1287          XOR A
E9AD CD D6 E0     1288          CALL OUT
E9B0 3E 40        1289          LD A,40H
E9B2 CD D6 E0     1290          CALL OUT
E9B5 2A 0A FF     1291          LD HL,(SYSTEM+0AH)
E9B8 7C           1292          LD A,H
E9B9 B5           1293          OR L
E9BA C0           1294          RET NZ
E9BB 3E 03        1295          LD A,3                     ;STOP INT (CTC)
E9BD D3 47        1296          OUT (CTC3),A
E9BF C9           1297          RET
E9C0              1298
E9C0              1299          ;*CURV ADR
E9C0 F5           1300  CURVADR PUSH AF
E9C1 C6 C0        1301          ADD A,0C0H
E9C3 67           1302          LD H,A
E9C4 2E 00        1303          LD L,0
E9C6 F1           1304          POP AF
E9C7 C9           1305          RET
E9C8              1306
E9C8              1307
E9C8              1308          ;***** KEY OFF *****
E9C8 F5           1309  KEYOFF  PUSH AF
E9C9 E5           1310          PUSH HL
E9CA D5           1311          PUSH DE
E9CB C5           1312          PUSH BC
E9CC 4F           1313          LD C,A
E9CD 06 00        1314          LD B,0
E9CF 2A 14 FF     1315          LD HL,(SYSTEM+14H)
E9D2 09           1316          ADD HL,BC
E9D3 7E           1317          LD A,(HL)
E9D4 5F           1318          LD E,A
E9D5 3D           1319          DEC A
E9D6 E6 7F        1320          AND 7FH
E9D8 FE 7F        1321          CP 7FH
E9DA 20 01        1322          JR NZ,SKIPX
E9DC AF           1323          XOR A
E9DD 57           1324  SKIPX   LD D,A
E9DE 7B           1325          LD A,E
E9DF E6 80        1326          AND 80H
E9E1 B2           1327          OR D
E9E2 77           1328          LD (HL),A
E9E3 E6 7F        1329          AND 7FH
E9E5 20 37        1330          JR NZ,KOFFNE2
E9E7 21 00 FF     1331          LD HL,SYSTEM
E9EA 3A 1A FF     1332          LD A,(SYSTEM+1AH)
E9ED 6F           1333          LD L,A
E9EE 7E           1334          LD A,(HL)
E9EF 3D           1335          DEC A
E9F0 77           1336          LD (HL),A
E9F1 20 19        1337          JR NZ,NOTEOFF
E9F3 3A 23 EA     1338          LD A,(ALLOFF)
E9F6 B7           1339          OR A
E9F7 28 13        1340          JR Z,NOTEOFF
E9F9 3A 0C FF     1341          LD A,(SYSTEM+0CH)          ;ALL NOTE OFF
E9FC F6 B0        1342          OR 0B0H
E9FE CD D6 E0     1343          CALL OUT
EA01 3E 7B        1344          LD A,7BH
```

▶ OS-9で起動したX68000がフロントスイッチ一発でいきなり電源が切れるのには心の臓を約1秒間停止させるほどの迫力があります。これは明らかにHuman68kからの退化だと思う。

西　宏通（21）沖縄県

```
EA03 CD D6 E0       1345          CALL OUT
EA06 AF             1346          XOR A
EA07 CD D6 E0       1347          CALL OUT
EA0A 18 12          1348          JR KOFFNE1
EA0C 3A 0C FF       1349  NOTEOFF LD A,(SYSTEM+0CH)       ;NOTE OFF
EA0F F6 80          1350          OR 080H
EA11 CD D6 E0       1351          CALL OUT
EA14 79             1352          LD A,C
EA15 CD D6 E0       1353          CALL OUT
EA18 AF             1354          XOR A
EA19 CD D6 E0       1355          CALL OUT
EA1C 18 00          1356          JR KOFFNE1
EA1E                1357  KOFFNE2 ;CALL RETRIGER
EA1E C1             1358  KOFFNE1 POP BC
EA1F D1             1359          POP DE
EA20 E1             1360          POP HL
EA21 F1             1361          POP AF
EA22 C9             1362          RET
EA23                1363
EA23 FF             1364  ALLOFF  DB 0FFH                 ;ALL NOTE OFF ヲ ツカワナイトキハ
 00H ニ スル
EA24                1365
EA24                1366          ;***** KEY ON *****
EA24 E5             1367  ?ENABLE PUSH HL
EA25 21 1E FF       1368          LD HL,SYSTEM+1EH
EA28 3A 10 FF       1369          LD A,(SYSTEM+10H)
EA2B 85             1370          ADD A,L
EA2C 6F             1371          LD L,A
EA2D 6E             1372          LD L,(HL)
EA2E 3A 11 FF       1373          LD A,(SYSTEM+11H)
EA31 2F             1374          CPL
EA32 A5             1375          AND L
EA33 E1             1376          POP HL
EA34 C9             1377          RET
EA35                1378
EA35 CD CA EA       1379  KEYON   CALL KEYX
EA38 4F             1380          LD C,A
EA39 DD 7E 3A       1381          LD A,(IX+3AH)
EA3C B7             1382          OR A
EA3D 28 06          1383          JR Z,KONNE1
EA3F DD 36 3A 00    1384          LD (IX+3AH),0
EA43 18 5B          1385          JR KONNE2         ;NEED NOT KEYON
EA45 EB             1386  KONNE1  EX DE,HL
EA46 CD B5 EA       1387          CALL SEARCH
EA49 38 46          1388          JR C,KONNE3       ;OVER SOUND
EA4B 71             1389          LD (HL),C         ;KEY
EA4C 23             1390          INC HL
EA4D 1A             1391          LD A,(DE)         ;LONG
EA4E CD 1D EB       1392          CALL LONGX
EA51 77             1393          LD (HL),A
EA52 06 00          1394          LD B,0
EA54 2A 14 FF       1395          LD HL,(SYSTEM+14H)
EA57 09             1396          ADD HL,BC
EA58 7E             1397          LD A,(HL)
EA59 3C             1398          INC A
EA5A F6 80          1399          OR 80H
EA5C 77             1400          LD (HL),A
EA5D E6 7F          1401          AND 7FH
EA5F 3D             1402          DEC A
EA60 20 29          1403          JR NZ,KONNE4
EA62 21 00 FF       1404          LD HL,SYSTEM
EA65 3A 1A FF       1405          LD A,(SYSTEM+1AH)
EA68 6F             1406          LD L,A
EA69 34             1407          INC (HL)
EA6A CD 24 EA       1408          CALL ?ENABLE
EA6D 28 22          1409          JR Z,KONNE3
EA6F 3A 0C FF       1410          LD A,(SYSTEM+0CH)       ;MIDI KEYON OUT
EA72 F6 90          1411          OR 090H
EA74 CD D6 E0       1412          CALL OUT
EA77 79             1413          LD A,C
EA78 CD D6 E0       1414          CALL OUT
EA7B DD 4E 3B       1415          LD C,(IX+3BH)
EA7E 0C             1416          INC C
EA7F 06 00          1417          LD B,0
EA81 2A 16 FF       1418          LD HL,(SYSTEM+16H)
EA84 09             1419          ADD HL,BC
EA85 7E             1420          LD A,(HL)
EA86 CD D6 E0       1421          CALL OUT
EA89 18 06          1422          JR KONNE3
EA8B CD 24 EA       1423  KONNE4  CALL ?ENABLE
EA8E C4 5C EB       1424          CALL NZ,RETRIGER
EA91 13             1425  KONNE3  INC DE
EA92 ED 53 12 FF    1426          LD (SYSTEM+12H),DE
EA96 AF             1427          XOR A
EA97 DD 77 4E       1428          LD (IX+4EH),A
EA9A DD 77 3B       1429          LD (IX+3BH),A
EA9D C3 92 E4       1430          JP LOOP
EAA0                1431
EAA0 EB             1432  KONNE2  EX DE,HL
EAA1 CD B5 EA       1433          CALL SEARCH
EAA4 38 09          1434          JR C,OVERX
EAA6 71             1435          LD (HL),C
EAA7 23             1436          INC HL
EAA8 1A             1437          LD A,(DE)
EAA9 CD 1D EB       1438          CALL LONGX
EAAC 77             1439          LD (HL),A
EAAD 18 E2          1440          JR KONNE3
EAAF                1441
EAAF 79             1442  OVERX   LD A,C
EAB0 CD C8 E9       1443          CALL KEYOFF
EAB3 18 DC          1444          JR KONNE3
EAB5                1445
EAB5 C5             1446  SEARCH  PUSH BC
EAB6 2A 18 FF       1447          LD HL,(SYSTEM+18H)
EAB9 23             1448          INC HL
EABA 06 10          1449          LD B,16
EABC 7E             1450  SEALP   LD A,(HL)
EABD B7             1451          OR A
EABE 28 07          1452          JR Z,FOUND
EAC0 23             1453          INC HL
EAC1 23             1454          INC HL
EAC2 10 F8          1455          DJNZ SEALP
EAC4 37             1456          SCF
EAC5 C1             1457          POP BC
EAC6 C9             1458          RET
EAC7 2B             1459  FOUND   DEC HL
EAC8 C1             1460          POP BC
EAC9 C9             1461          RET
EACA                1462
EACA E5             1463  KEYX    PUSH HL
EACB D5             1464          PUSH DE
EACC C5             1465          PUSH BC
EACD F5             1466          PUSH AF
EACE 6F             1467          LD L,A
EACF 26 00          1468          LD H,0
EAD1 11 0C 00       1469          LD DE,12
EAD4 CD 53 01       1470          CALL DIV
EAD7 2A 16 FF       1471          LD HL,(SYSTEM+16H)
EADA 19             1472          ADD HL,DE
EADB 11 40 00       1473          LD DE,40H
EADE 19             1474          ADD HL,DE
EADF DD 7E 4D       1475          LD A,(IX+4DH)
EAE2 B7             1476          OR A
EAE3 28 07          1477          JR Z,KEYXNE
EAE5 D6 80          1478          SUB 80H
EAE7 77             1479          LD (HL),A
EAE8 DD 36 4D 00    1480          LD (IX+4DH),0
EAEC 46             1481  KEYXNE  LD B,(HL)
EAED F1             1482          POP AF
EAEE 80             1483          ADD A,B
EAEF DD 86 7E       1484          ADD A,(IX+7EH)          ;KEY SHIFT (PART)
EAF2 47             1485          LD B,A
EAF3 DD 7E 7F       1486          LD A,(IX+7FH)           ;SHIFT MODE
EAF6 4F             1487          LD C,A
EAF7 B7             1488          OR A
EAF8 28 03          1489          JR Z,KSCSK1
EAFA AF             1490          XOR A
EAFB 18 03          1491          JR KSCSK2
EAFD 3A 01 FF       1492  KSCSK1  LD A,(SYSTEM+1)         ;KEY SHIFT (TOTAL)
EB00 80             1493  KSCSK2  ADD A,B
EB01 0D             1494          DEC C
EB02 0D             1495          DEC C
EB03 28 08          1496          JR Z,PAT1
EB05 0D             1497          DEC C
EB06 20 0F          1498          JR NZ,KSCSK3
EB08 21 80 ED       1499          LD HL,PATCH2
EB0B 18 03          1500          JR KSCSK4
EB0D 21 00 ED       1501  PAT1    LD HL,PATCH1
EB10 E6 7F          1502  KSCSK4  AND 7FH
EB12 4F             1503          LD C,A
EB13 06 00          1504          LD B,0
EB15 09             1505          ADD HL,BC
EB16 7E             1506          LD A,(HL)
EB17 E6 7F          1507  KSCSK3  AND 7FH
EB19 C1             1508          POP BC
EB1A D1             1509          POP DE
EB1B E1             1510          POP HL
EB1C C9             1511          RET
EB1D                1512
EB1D E5             1513  LONGX   PUSH HL
EB1E D5             1514          PUSH DE
EB1F C5             1515          PUSH BC
EB20 47             1516          LD B,A
EB21 DD 7E 4E       1517          LD A,(IX+4EH)
EB24 B7             1518          OR A
EB25 28 11          1519          JR Z,BGATE
EB27 3D             1520          DEC A
EB28 28 09          1521          JR Z,BSTACC
EB2A 3D             1522          DEC A
EB2B 28 03          1523          JR Z,BTENU
EB2D AF             1524          XOR A
EB2E 18 28          1525          JR ELONGX
EB30 78             1526  BTENU   LD A,B
EB31 18 25          1527          JR ELONGX
EB33 DD 7E 3C       1528  BSTACC  LD A,(IX+3CH)
EB36 18 03          1529          JR BGATE2
EB38 DD 7E 20       1530  BGATE   LD A,(IX+20H)
EB3B E6 07          1531  BGATE2  AND 07H
EB3D 3C             1532          INC A
EB3E 58             1533          LD E,B
EB3F 16 00          1534          LD D,0
EB41 6A             1535          LD L,D
EB42 62             1536          LD H,D
EB43 47             1537          LD B,A
EB44 19             1538  BGLOOP  ADD HL,DE
EB45 10 FD          1539          DJNZ BGLOOP
EB47 CB 3C          1540          SRL H
EB49 CB 1D          1541          RR L
EB4B CB 3C          1542          SRL H
EB4D CB 1D          1543          RR L
EB4F CB 3C          1544          SRL H
EB51 CB 1D          1545          RR L
EB53 7D             1546          LD A,L
EB54 B7             1547          OR A
EB55 20 01          1548          JR NZ,ELONGX
EB57 3C             1549          INC A
EB58 C1             1550  ELONGX  POP BC
EB59 D1             1551          POP DE
EB5A E1             1552          POP HL
EB5B C9             1553          RET
EB5C                1554
EB5C                1555  RETRIGER
EB5C 3A 0C FF       1556          LD A,(SYSTEM+0CH)
EB5F F6 80          1557          OR 80H
EB61 47             1558          LD B,A
EB62 CD D6 E0       1559          CALL OUT
EB65 79             1560          LD A,C
EB66 CD D6 E0       1561          CALL OUT
EB69 AF             1562          XOR A
EB6A CD D6 E0       1563          CALL OUT
EB6D 78             1564          LD A,B
EB6E C6 10          1565          ADD A,10H
EB70 CD D6 E0       1566          CALL OUT
EB73 79             1567          LD A,C
EB74 CD D6 E0       1568          CALL OUT
EB77 DD 4E 3B       1569          LD C,(IX+3BH)
EB7A 0C             1570          INC C
EB7B 06 00          1571          LD B,0
EB7D 2A 16 FF       1572          LD HL,(SYSTEM+16H)
EB80 09             1573          ADD HL,BC
EB81 7E             1574          LD A,(HL)
EB82 CD D6 E0       1575          CALL OUT
EB85 C9             1576          RET
EB86                1577
EB86 F5             1578  VOLSET  PUSH AF
EB87 3A 0C FF       1579          LD A,(SYSTEM+0CH)
EB8A F6 B0          1580          OR 0B0H
EB8C CD D6 E0       1581          CALL OUT
EB8F 3E 07          1582          LD A,07H
EB91 CD D6 E0       1583          CALL OUT
EB94 F1             1584          POP AF
EB95 DD 77 04       1585          LD (IX+04H),A
EB98 CD D6 E0       1586          CALL OUT
EB9B C9             1587          RET
EB9C                1588
EB9C                1589          ;***** SET *****
EB9C 1A             1590  PPSET   LD A,(DE)
EB9D E6 7F          1591          AND 7FH
EB9F DD 77 08       1592          LD (IX+08H),A
EBA2 C3 F6 E7       1593          JP UP1
EBA5                1594
EBA5 1A             1595  PSET    LD A,(DE)
EBA6 E6 7F          1596          AND 7FH
EBA8 DD 77 09       1597          LD (IX+09H),A
EBAB C3 F6 E7       1598          JP UP1
EBAE                1599
EBAE 1A             1600  MPSET   LD A,(DE)
EBAF E6 7F          1601          AND 7FH
EBB1 DD 77 0A       1602          LD (IX+0AH),A
EBB4 C3 F6 E7       1603          JP UP1
EBB7                1604
EBB7 1A             1605  MFSET   LD A,(DE)
EBB8 E6 7F          1606          AND 7FH
EBBA DD 77 0B       1607          LD (IX+0BH),A
EBBD C3 F6 E7       1608          JP UP1
EBC0                1609
EBC0 1A             1610  FSET    LD A,(DE)
EBC1 E6 7F          1611          AND 7FH
EBC3 DD 77 0C       1612          LD (IX+0CH),A
EBC6 C3 F6 E7       1613          JP UP1
EBC9                1614
EBC9 1A             1615  FFSET   LD A,(DE)
EBCA E6 7F          1616          AND 7FH
EBCC DD 77 0D       1617          LD (IX+0DH),A
EBCF C3 F6 E7       1618          JP UP1
EBD2                1619
EBD2 1A             1620  NSET    LD A,(DE)
EBD3 E6 7F          1621          AND 7FH
EBD5 DD 77 01       1622          LD (IX+01H),A
EBD8 C3 F6 E7       1623          JP UP1
EBDB                1624
EBDB 1A             1625  A1SET   LD A,(DE)
EBDC E6 7F          1626          AND 7FH
EBDE DD 77 02       1627          LD (IX+02H),A
```

▶ついに（で）さんの顔写真が出たー。やはり思ったとおりだったなあ。新連載がどんなものになるか楽しみだ（しかしちょっと怖い）。これにて1杯ごちそうさま。

吉見　考正（19）静岡県

```
EBE1 C3 F6 E7      1628          JP UP1
EBE4               1629
EBE4 1A            1630  A2SET   LD A,(DE)
EBE5 E6 7F         1631          AND 7FH
EBE7 DD 77 03      1632          LD (IX+03H),A
EBEA C3 F6 E7      1633          JP UP1
EBED               1634
EBED 1A            1635  CHSET   LD A,(DE)
EBEE E6 0F         1636          AND 0FH
EBF0 DD 77 00      1637          LD (IX+00H),A
EBF3 D5            1638          PUSH DE
EBF4 CD D0 E2      1639          CALL INITSET
EBF7 D1            1640          POP DE
EBF8 C3 F6 E7      1641          JP UP1
EBFB               1642
EBFB 1A            1643  STSET   LD A,(DE)
EBFC E6 07         1644          AND 07H
EBFE DD 77 3C      1645          LD (IX+3CH),A
EC01 C3 F6 E7      1646          JP UP1
EC04               1647
EC04               1648          ;***** MIDI OUT *****
EC04 3E 01         1649  MIDIOUT LD A,1
EC06 32 6E E1      1650          LD (EXF),A
EC09 1A            1651          LD A,(DE)
EC0A 13            1652          INC DE
EC0B B7            1653          OR A
EC0C 28 08         1654          JR Z,MONE1
EC0E 47            1655          LD B,A
EC0F 1A            1656  MOLOOP  LD A,(DE)
EC10 13            1657          INC DE
EC11 CD D6 E0      1658          CALL OUT
EC14 10 F9         1659          DJNZ MOLOOP
EC16 ED 53 12 FF   1660  MONE1   LD (SYSTEM+12H),DE
EC1A AF            1661          XOR A
EC1B 32 6E E1      1662          LD (EXF),A
EC1E C3 92 E4      1663          JP LOOP
EC21               1664
EC21               1665          ;***** KEY SHIFT SET *****
EC21 1A            1666  SHIFT   LD A,(DE)
EC22 DD 77 7E      1667          LD (IX+7EH),A
EC25 C3 F6 E7      1668          JP UP1
EC28               1669
EC28               1670          ;***** FELMATA (0E6H WAIT2) *****
EC28 EB            1671  FEL     EX DE,HL
EC29 5E            1672          LD E,(HL)
EC2A 23            1673          INC HL
EC2B 56            1674          LD D,(HL)
EC2C EB            1675          EX DE,HL
EC2D 22 1C FF      1676          LD (SYSTEM+1CH),HL
EC30 C3 F6 E7      1677          JP UP1
EC33               1678
EC33               1679          ;***** CONTROL CHANGE (0E7H) *****
EC33 3A 0C FF      1680  CHANGE  LD A,(SYSTEM+0CH)
EC36 F6 B0         1681          OR 0B0H
EC38 CD D6 E0      1682          CALL OUT
EC3B 1A            1683          LD A,(DE)
EC3C CD D6 E0      1684          CALL OUT
EC3F 13            1685          INC DE
EC40 1A            1686          LD A,(DE)
EC41 CD D6 E0      1687          CALL OUT
EC44 C3 F6 E7      1688          JP UP1
EC47               1689
EC47               1690          ;***** SHIFT MODE SET *****
EC47 1A            1691  SMODE   LD A,(DE)
EC48 DD 77 7F      1692          LD (IX+7FH),A
EC4B C3 F6 E7      1693          JP UP1
EC4E               1694
EC4E               1695          ;***** CURV PATTERN ADR. *****
```

```
EC4E FD 4E 04      1696  CURSUB  LD C,(IY+4)
EC51 FD 46 05      1697          LD B,(IY+5)
EC54 FD 6E 06      1698          LD L,(IY+6)
EC57 FD 66 07      1699          LD H,(IY+7)
EC5A FD 7E 00      1700          LD A,(IY)
EC5D E6 04         1701          AND 4
EC5F 20 17         1702          JR NZ,YTYPE
EC61 09            1703          ADD HL,BC
EC62 FD 75 06      1704          LD (IY+6),L
EC65 FD 74 07      1705          LD (IY+7),H
EC68 D0            1706          RET NC
EC69 FD 6E 02      1707          LD L,(IY+2)
EC6C FD 66 03      1708          LD H,(IY+3)
EC6F 23            1709          INC HL
EC70 FD 75 02      1710          LD (IY+2),L
EC73 FD 74 03      1711          LD (IY+3),H
EC76 37            1712          SCF
EC77 C9            1713          RET
EC78 09            1714  YTYPE   ADD HL,BC
EC79 FD 75 06      1715          LD (IY+6),L
EC7C FD 74 07      1716          LD (IY+7),H
EC7F 4C            1717          LD C,H
EC80 06 00         1718          LD B,0
EC82 FD 6E 02      1719          LD L,(IY+2)
EC85 FD 66 03      1720          LD H,(IY+3)
EC88 09            1721          ADD HL,BC
EC89 37            1722          SCF
EC8A C9            1723          RET
EC8B               1724
EC8B FD 77 01      1725  RTDSET  LD (IY+01H),A
EC8E 13            1726          INC DE
EC8F 1A            1727          LD A,(DE)
EC90 6F            1728          LD L,A
EC91 FD 75 08      1729          LD (IY+08H),L
EC94 13            1730          INC DE
EC95 1A            1731          LD A,(DE)
EC96 67            1732          LD H,A
EC97 FD 74 09      1733          LD (IY+09H),H
EC9A B7            1734          OR A
EC9B 28 1C         1735          JR Z,RTDYTP
EC9D D5            1736          PUSH DE
EC9E 3E FF         1737          LD A,255
ECA0 01 00 00      1738          LD BC,0
ECA3 EB            1739          EX DE,HL
ECA4 CD 70 E1      1740          CALL DIV24
ECA7 FD 71 04      1741          LD (IY+04H),C
ECAA FD 70 05      1742          LD (IY+05H),B
ECAD CB 38         1743          SRL B
ECAF CB 19         1744          RR C
ECB1 FD 71 06      1745          LD (IY+06H),C
ECB4 FD 70 07      1746          LD (IY+07H),B
ECB7 D1            1747          POP DE
ECB8 C9            1748          RET
ECB9 D5            1749  RTDYTP  PUSH DE
ECBA EB            1750          EX DE,HL
ECBB 21 00 FF      1751          LD HL,0FF00H
ECBE CD 53 01      1752          CALL DIV
ECC1 FD 75 04      1753          LD (IY+04H),L
ECC4 FD 74 05      1754          LD (IY+05H),H
ECC7 AF            1755          XOR A
ECC8 FD 77 06      1756          LD (IY+06H),A
ECCB FD 77 07      1757          LD (IY+07H),A
ECCE 3E 04         1758          LD A,4
ECD0 FD B6 00      1759          OR (IY+00H)
ECD3 FD 77 00      1760          LD (IY+00H),A
ECD6 D1            1761          POP DE
ECD7 C9            1762          RET
```

リスト6 MACROソースリスト

```
0000                  1          ;***** MUSIC MACRO *****
D000                  2          ORG 0D000H
D000                  3
0139 P                4  MUL     EQU 0139H              ;HL*DE->HL
0153 P                5  DIV     EQU 0153H              ;HL/DE->HL...DE
E0D6 P                6  #OUT    EQU 0E0D6H
E154 P                7  #ALLOUT EQU 0E154H
E296 P                8  #STEMPO EQU 0E296H
E16E P                9  #EXF    EQU 0E16EH
0981 P               10  #DC1    EQU 0981H
E16C P               11  #DC2    EQU 0E16CH
2030 P               12  DTSTART EQU 02030H
FF30 P               13  OPENF   EQU 0FF30H
FF31 P               14  USEMAPC EQU 0FF31H
FF32 P               15  USEMAP  EQU 0FF32H
FF52 P               16  TRKMAP  EQU 0FF52H
FF62 P               17  TRKADR  EQU 0FF62H
EE00 P               18  TRKWORK EQU 0EE00H
0090 P               19  TLEN    EQU 090H
0010 P               20  TRK     EQU 16
0800 P               21  MIDIX   EQU 80H*16
0030 P               22  SYSLEN  EQU 30H
FF92 P               23  TRKNO   EQU 0FF92H
FF82 P               24  MIDICH  EQU 0FF82H
FFA0 P               25  RTBUFF  EQU 0FFA0H
F700 P               26  MIDIKEY EQU 0F700H
0047 P               27  CTC3    EQU 047H
0043 P               28  SIOC    EQU 043H
FF93 P               29  SHIFT1  EQU 0FF93H
D000                 30
D000 C3 15 D0        31  SUB0    JP MACROST
D003 C3 C6 D8        32  SUB1    JP OPEN
D006 C3 38 D9        33  SUB2    JP NEWTRACK
D009 C3 FE D9        34  SUB3    JP CLOSE
D00C C3 4C DA        35  SUB4    JP PLAY
D00F C3 01 DC        36  SUB5    JP CONT
D012 C3 17 DC        37  SUB6    JP STOP
D015                 38
D015                 39
D015 F5              40  MACROST PUSH AF
D016 78              41          LD A,B
D017 FE 03           42          CP 3
D019 C2 47 DA        43          JP NZ,ER4X             ;Type mismatch error
D01C 3A 30 FF        44          LD A,(OPENF)
D01F FE 55           45          CP 55H
D021 20 5A           46          JR NZ,ER44             ;Not open error
D023 C5              47          PUSH BC
D024 D5              48          PUSH DE
D025 E5              49          PUSH HL
D026 C5              50          PUSH BC
D027 21 AD D8        51          LD HL,MABUFF
D02A 01 B5 05        52          LD BC,05B5H
D02D 3E 01           53          LD A,1
D02F F3              54          DI
D030 D3 B4           55          OUT (0B4H),A
D032 ED B2           56          INIR
D034 3A 92 FF        57          LD A,(TRKNO)   ;TRACK NO.(0-15)
D037 4F              58          LD C,A
D038 06 00           59          LD B,0
D03A 21 52 FF        60          LD HL,TRKMAP
D03D 09              61          ADD HL,BC
D03E 4E              62          LD C,(HL)
D03F 21 32 FF        63          LD HL,USEMAP
D042 09              64          ADD HL,BC
```

```
D043 01 B5 05        65          LD BC,05B5H
D046 3E 01           66          LD A,1
D048 D3 B4           67          OUT (0B4H),A
D04A ED B3           68          OTIR
D04C FB              69          EI
D04D C1              70          POP BC
D04E E1              71          POP HL
D04F E5              72          PUSH HL
D050 ED 73 A1 D8     73          LD (SPBUFF),SP
D054 CD D6 D0        74          CALL MACRO
D057 ED 7B A1 D8     75  RETAD   LD SP,(SPBUFF)
D05B F5              76          PUSH AF
D05C 21 AD D8        77          LD HL,MABUFF
D05F 01 B5 05        78          LD BC,05B5H
D062 3E 01           79          LD A,01H
D064 F3              80          DI
D065 D3 B4           81          OUT (0B4H),A
D067 ED B3           82          OTIR
D069 FB              83          EI
D06A F1              84          POP AF
D06B E1              85          POP HL
D06C ED 5B 9B D8     86          LD DE,(OADR)
D070 73              87          LD (HL),E
D071 23              88          INC HL
D072 72              89          LD (HL),D
D073 2B              90          DEC HL
D074 D1              91          POP DE
D075 C1              92          POP BC
D076 06 02           93          LD B,2
D078 B7              94          OR A
D079 20 04           95          JR NZ,ERROR
D07B F1              96          POP AF
D07C C9              97          RET
D07D                 98
D07D 3E 2C           99  ER44    LD A,44
D07F 33             100  ERROR   INC SP
D080 33             101          INC SP
D081 DD E9          102          JP (IX)
D083                103
D083                104
D083 3A B5 D8       105  NEWMAP  LD A,(NEWF)
D086 C6 20          106          ADD A,20H
D088 32 B5 D8       107          LD (NEWF),A
D08B 3A 31 FF       108          LD A,(USEMAPC)
D08E FE 20          109          CP 20H
D090 D2 16 D1       110          JP NC,ER6
D093 CD B9 D0       111          CALL NEWSEA
D096 DA 16 D1       112          JP C,ER6
D099 47             113          LD B,A
D09A 3A 31 FF       114          LD A,(USEMAPC)
D09D 3C             115          INC A
D09E 32 31 FF       116          LD (USEMAPC),A
D0A1 5F             117          LD E,A
D0A2 16 00          118          LD D,0
D0A4 21 31 FF       119          LD HL,USEMAPC
D0A7 19             120          ADD HL,DE
D0A8 70             121          LD (HL),B
D0A9 3A B5 D8       122          LD A,(NEWF)
D0AC 07             123          RLCA
D0AD 07             124          RLCA
D0AE 07             125          RLCA
D0AF E6 07          126          AND 07H
D0B1 F3             127          DI
D0B2 D3 B4          128          OUT (0B4H),A
```

▶ "満開の電子ちゃん"，これはなかなか笑える！ 連載広告にしたらどうでしょうか？
五百住 正樹 (19) 大阪府

```
D0B4 78              129            LD A,B
D0B5 D3 B5           130            OUT (0B5H),A
D0B7 FB              131            EI
D0B8 C9              132            RET
D0B9                 133
D0B9 E5              134   NEWSEA   PUSH HL
D0BA C5              135            PUSH BC
D0BB 21 6F 05        136            LD HL,056FH
D0BE 06 30           137            LD B,30H
D0C0 0E 2F           138            LD C,2FH
D0C2 7E              139   NEWSLP   LD A,(HL)
D0C3 E6 80           140            AND 80H
D0C5 28 08           141            JR Z,NESFIND
D0C7 2B              142            DEC HL
D0C8 0D              143            DEC C
D0C9 10 F7           144            DJNZ NEWSLP
D0CB 37              145            SCF
D0CC C1              146            POP BC
D0CD E1              147            POP HL
D0CE C9              148            RET
D0CF CB FE           149   NESFIND  SET 7,(HL)
D0D1 B7              150            OR A
D0D2 79              151            LD A,C
D0D3 C1              152            POP BC
D0D4 E1              153            POP HL
D0D5 C9              154            RET
D0D6                 155
D0D6 78              156   MACRO    LD A,B
D0D7 FE 03           157            CP 3
D0D9 3E 04           158            LD A,4
D0DB C0              159            RET NZ
D0DC 7E              160            LD A,(HL)
D0DD B7              161            OR A
D0DE C8              162            RET Z
D0DF 23              163            INC HL
D0E0 5E              164            LD E,(HL)
D0E1 23              165            INC HL
D0E2 56              166            LD D,(HL)
D0E3 ED 53 9D D8     167            LD (SADR),DE
D0E7 6F              168            LD L,A
D0E8 26 00           169            LD H,0
D0EA 19              170            ADD HL,DE
D0EB 36 00           171            LD (HL),0
D0ED 2B              172            DEC HL
D0EE 22 9F D8        173            LD (SEADR),HL
D0F1                 174
D0F1 3A 9C D8        175   MALOOP   LD A,(OADR+1)
D0F4 FE BE           176            CP 0BEH
D0F6 30 1E           177            JR NC,ER6
D0F8 47              178            LD B,A
D0F9 3A B5 D8        179            LD A,(NEWF)
D0FC B8              180            CP B
D0FD DC 83 D0        181            CALL C,NEWMAP
D100 2A 9F D8        182            LD HL,(SEADR)
D103 ED 5B 9D D8     183            LD DE,(SADR)
D107 CD 55 DC        184            CALL SPSKIP
D10A B7              185            OR A
D10B ED 52           186            SBC HL,DE
D10D 30 1A           187            JR NC,COMX
D10F 2A 9B D8        188            LD HL,(OADR)
D112 36 FE           189            LD (HL),0FEH
D114 AF              190            XOR A
D115 C9              191            RET
D116                 192
D116 3E 06           193   ER6      LD A,6
D118 C3 57 D0        194            JP RETAD
D11B                 195
D11B                 196
D11B FE 61           197   UTOL     CP 61H
D11D D8              198            RET C
D11E FE 7B           199            CP 7BH
D120 D0              200            RET NC
D121 D6 20           201            SUB 20H
D123 C9              202            RET
D124                 203
D124 3E 03           204   ER3      LD A,3
D126 C3 57 D0        205            JP RETAD
D129                 206
D129                 207            ;***** COMMAND CHECK *****
D129 1A              208   COMX     LD A,(DE)
D12A 13              209            INC DE
D12B FE 40           210            CP "@"
D12D CA A8 D5        211            JP Z,EXP1
D130 FE 5B           212            CP "["
D132 CA 3A D7        213            JP Z,EXP2
D135 FE 3F           214            CP "?"
D137 CA 7F D5        215            JP Z,EXP3
D13A CD 1B D1        216            CALL UTOL
D13D FE 41           217            CP 41H
D13F 38 05           218            JR C,BCNEX
D141 FE 48           219            CP 48H
D143 DA 83 D3        220            JP C,OMPU
D146 4F              221   BCNEX    LD C,A
D147 21 5E D1        222            LD HL,BCOM
D14A 7E              223   BCOMLP   LD A,(HL)
D14B B7              224            OR A
D14C 28 D6           225            JR Z,ER3
D14E 23              226            INC HL
D14F B9              227            CP C
D150 28 04           228            JR Z,BCFIND
D152 23              229            INC HL
D153 23              230            INC HL
D154 18 F4           231            JR BCOMLP
D156 4E              232   BCFIND   LD C,(HL)
D157 23              233            INC HL
D158 66              234            LD H,(HL)
D159 69              235            LD L,C
D15A CD 55 DC        236            CALL SPSKIP
D15D E9              237            JP (HL)
D15E                 238
D15E 2A              239   BCOM     DM "*"
D15F C3 D1           240            DW DSHARP
D161 2B              241            DM "+"
D162 B5 D1           242            DW SHARP?
D164 2D              243            DM "-"
D165 CB D1           244            DW FLAT?
D167 52              245            DM "R"
D168 FD D1           246            DW REST
D16A 4E              247            DM "N"
D16B 05 D2           248            DW OMPU2
D16D 51              249            DM "Q"
D16E 15 D2           250            DW GATE
D170 24              251            DM "$"
D171 2E D2           252            DW TIEOUT
D173 26              253            DM "&"
D174 36 D2           254            DW TIEIN
D176 4C              255            DM "L"
D177 71 D2           256            DW LENGTH
D179 54              257            DM "T"
D17A 7A D2           258            DW TEMPO
D17C 4F              259            DM "O"
D17D 92 D2           260            DW OCT
D17F 3E              261            DM ">"
D180 A6 D2           262            DW OCTUP
D182 3C              263            DM "<"
D183 B5 D2           264            DW OCTDOWN
D185 59              265            DM "Y"
D186 C3 D2           266            DW MIDIOUT
D188 27              267            DM "'"
D189 FD D2           268            DW STACC
D18B 5F              269            DM "_"
D18C 1E D3           270            DW TENU
D18E 5E              271            DM "^"
D18F 26 D3           272            DW ACCENT
D191 3A              273            DM ":"
D192 4A D3           274            DW WAIT
D194 7C              275            DM "|"
D195 68 D3           276            DW BAR
D197 7B              277            DB 7BH
D198 70 D3           278            DW REPIN
D19A 7D              279            DB 7DH
D19B 7B D3           280            DW REPOUT
D19D 28              281            DM "("
D19E 24 D5           282            DW KAKKO
D1A0 57              283            DM "W"
D1A1 EC D1           284            DW FEL
D1A3 4D              285            DM "M"
D1A4 44 D5           286            DW SETCH
D1A6 56              287            DM "V"
D1A7 5D D5           288            DW VOLUME
D1A9 00              289            DB 0
D1AA                 290
D1AA 23              291   UP0      INC HL
D1AB 22 9B D8        292            LD (OADR),HL
D1AE ED 53 9D D8     293   UPX0     LD (SADR),DE
D1B2 C3 F1 D0        294            JP MALOOP
D1B5                 295
D1B5                 296
D1B5 1A              297   SHARP?   LD A,(DE)
D1B6 FE 2B           298            CP "+"
D1B8 28 08           299            JR Z,DSHARP2
D1BA FE 2D           300            CP "-"
D1BC 28 09           301            JR Z,NATU
D1BE 3E 01           302            LD A,1
D1C0 18 15           303            JR RINGI
D1C2 13              304   DSHARP2  INC DE
D1C3 3E 02           305   DSHARP   LD A,2
D1C5 18 10           306            JR RINGI
D1C7 13              307   NATU     INC DE
D1C8 AF              308            XOR A
D1C9 18 0C           309            JR RINGI
D1CB 1A              310   FLAT?    LD A,(DE)
D1CC FE 2D           311            CP "-"
D1CE 28 04           312            JR Z,DFLAT
D1D0 3E FF           313            LD A,-1
D1D2 18 03           314            JR RINGI
D1D4 13              315   DFLAT    INC DE
D1D5 3E FE           316            LD A,-2
D1D7 ED 53 9D D8     317   RINGI    LD (SADR),DE
D1DB 2A 9B D8        318            LD HL,(OADR)
D1DE 36 E3           319            LD (HL),0E3H
D1E0 23              320            INC HL
D1E1 77              321            LD (HL),A
D1E2 3A BB D8        322            LD A,(KMFRAG)
D1E5 B7              323            OR A
D1E6 CA 24 D1        324            JP Z,ER3
D1E9 C3 AA D1        325            JP UP0
D1EC                 326
D1EC CD 68 D4        327   FEL      CALL LKAZU
D1EF 4D              328            LD C,L
D1F0 44              329            LD B,H
D1F1 2A 9B D8        330            LD HL,(OADR)
D1F4 36 E6           331            LD (HL),0E6H
D1F6 23              332            INC HL
D1F7 71              333            LD (HL),C
D1F8 23              334            INC HL
D1F9 70              335            LD (HL),B
D1FA C3 AA D1        336            JP UP0
D1FD                 337
D1FD 2A 9B D8        338   REST     LD HL,(OADR)
D200 36 EF           339            LD (HL),0EFH
D202 C3 E7 D3        340            JP OMPUL
D205                 341
D205 CD 37 D4        342   OMPU2    CALL DKAZU
D208 2A 9B D8        343            LD HL,(OADR)
D20B 77              344            LD (HL),A
D20C 23              345            INC HL
D20D 3A 9A D8        346            LD A,(LBUFF)
D210 77              347            LD (HL),A
D211 47              348            LD B,A
D212 C3 ED D3        349            JP OMPUL2
D215                 350
D215 CD 03 D4        351   GATE     CALL KAZU
D218 7C              352            LD A,H
D219 B7              353            OR A
D21A C2 24 D1        354            JP NZ,ER3
D21D 7D              355            LD A,L
D21E 3D              356            DEC A
D21F FE 08           357            CP 8
D221 D2 24 D1        358            JP NC,ER3
D224 2A 9B D8        359            LD HL,(OADR)
D227 36 90           360            LD (HL),90H
D229 23              361            INC HL
D22A 77              362            LD (HL),A
D22B C3 AA D1        363            JP UP0
D22E                 364
D22E 2A 9B D8        365   TIEOUT   LD HL,(OADR)
D231 36 8A           366            LD (HL),8AH
D233 C3 AA D1        367            JP UP0
D236                 368
D236 2A 9B D8        369   TIEIN    LD HL,(OADR)
D239 36 8B           370            LD (HL),8BH
D23B 3A B8 D8        371            LD A,(MODE)
D23E 47              372            LD B,A
D23F 3A B9 D8        373            LD A,(OMPF)
D242 A0              374            AND B
D243 CA AA D1        375            JP Z,UP0
D246 C5              376            PUSH BC
D247 D5              377            PUSH DE
D248 E5              378            PUSH HL
D249 ED 5B B6 D8     379            LD DE,(BOADR)
D24D B7              380            OR A
D24E ED 52           381            SBC HL,DE
D250 23              382            INC HL
D251 4D              383            LD C,L
D252 44              384            LD B,H
D253 E1              385            POP HL
D254 E5              386            PUSH HL
D255 5D              387            LD E,L
D256 54              388            LD D,H
D257 13              389            INC DE
D258 ED B8           390            LDDR
D25A 23              391            INC HL
D25B 36 8A           392            LD (HL),8AH
D25D 3A B9 D8        393            LD A,(OMPF)
D260 E6 7F           394            AND 7FH
D262 32 B9 D8        395            LD (OMPF),A
D265 3E FF           396            LD A,255
D267 32 BA D8        397            LD (TIEF),A
D26A E1              398            POP HL
D26B D1              399            POP DE
D26C C1              400            POP BC
D26D 23              401            INC HL
D26E C3 AA D1        402            JP UP0
D271                 403
D271 CD C4 D4        404   LENGTH   CALL LKAZU2
D274 32 9A D8        405            LD (LBUFF),A
D277 C3 AE D1        406            JP UPX0
```

▶ 学園祭に向けて，3Dのアニメーションを16㎜コマ撮りで制作しようと思っている最中に，こんな記事が載っているとは，もう最高！　うれしー！　西方　茂樹 (20) 茨城県

```
D27A                407
D27A CD 03 D4       408  TEMPO    CALL KAZU
D27D 7C             409           LD A,H
D27E B7             410           OR A
D27F C2 24 D1       411           JP NZ,ER3
D282 7D             412           LD A,L
D283 FE 1E          413           CP 30
D285 DA 24 D1       414           JP C,ER3
D288 2A 9B D8       415           LD HL,(OADR)
D28B 36 9E          416           LD (HL),9EH
D28D 23             417           INC HL
D28E 77             418           LD (HL),A
D28F C3 AA D1       419           JP UP0
D292                420
D292 CD 03 D4       421  OCT      CALL KAZU
D295 7C             422           LD A,H
D296 B7             423           OR A
D297 C2 24 D1       424           JP NZ,ER3
D29A 7D             425           LD A,L
D29B FE 0B          426           CP 11
D29D D2 24 D1       427           JP NC,ER3
D2A0 32 99 D8       428           LD (OBUFF),A
D2A3 C3 AE D1       429           JP UPX0
D2A6                430
D2A6 3A 99 D8       431  OCTUP    LD A,(OBUFF)
D2A9 FE 0A          432           CP 10
D2AB D2 24 D1       433           JP NC,ER3
D2AE 3C             434           INC A
D2AF 32 99 D8       435           LD (OBUFF),A
D2B2 C3 AE D1       436           JP UPX0
D2B5                437
D2B5 3A 99 D8       438  OCTDOWN LD A,(OBUFF)
D2B8 B7             439           OR A
D2B9 CA 24 D1       440           JP Z,ER3
D2BC 3D             441           DEC A
D2BD 32 99 D8       442           LD (OBUFF),A
D2C0 C3 AE D1       443           JP UPX0
D2C3                444
D2C3 1A             445  MIDIOUT LD A,(DE)
D2C4 FE 5B          446           CP "["
D2C6 C2 24 D1       447           JP NZ,ER3
D2C9 13             448           INC DE
D2CA 2A 9B D8       449           LD HL,(OADR)
D2CD 23             450           INC HL
D2CE 0E 00          451           LD C,0
D2D0 1A             452           LD A,(DE)
D2D1 FE 5D          453           CP "]"
D2D3 CA 24 D1       454           JP Z,ER3
D2D6 E5             455  OUTLP    PUSH HL
D2D7 CD 48 D4       456           CALL HKAZU
D2DA 45             457           LD B,L
D2DB 7C             458           LD A,H
D2DC E1             459           POP HL
D2DD B7             460           OR A
D2DE C2 24 D1       461           JP NZ,ER3
D2E1 23             462           INC HL
D2E2 0C             463           INC C
D2E3 70             464           LD (HL),B
D2E4 1A             465           LD A,(DE)
D2E5 13             466           INC DE
D2E6 FE 5D          467           CP "]"
D2E8 28 07          468           JR Z,OUTEND
D2EA FE 2C          469           CP ","
D2EC 28 E8          470           JR Z,OUTLP
D2EE C3 24 D1       471           JP ER3
D2F1 E5             472  OUTEND   PUSH HL
D2F2 2A 9B D8       473           LD HL,(OADR)
D2F5 36 E4          474           LD (HL),0E4H
D2F7 23             475           INC HL
D2F8 71             476           LD (HL),C
D2F9 E1             477           POP HL
D2FA C3 AA D1       478           JP UP0
D2FD                479
D2FD 2A 9B D8       480  STACC    LD HL,(OADR)
D300 1A             481           LD A,(DE)
D301 FE 3D          482           CP "="
D303 28 05          483           JR Z,STCSET
D305 36 88          484           LD (HL),88H
D307 C3 AA D1       485           JP UP0
D30A 13             486  STCSET   INC DE
D30B 2A 9B D8       487           LD HL,(OADR)
D30E 36 CA          488           LD (HL),0CAH
D310 CD 37 D4       489           CALL DKAZU
D313 23             490           INC HL
D314 3D             491           DEC A
D315 FE 08          492           CP 8
D317 D2 24 D1       493           JP NC,ER3
D31A 77             494           LD (HL),A
D31B C3 AA D1       495           JP UP0
D31E                496
D31E 2A 9B D8       497  TENU     LD HL,(OADR)
D321 36 89          498           LD (HL),89H
D323 C3 AA D1       499           JP UP0
D326                500
D326 2A 9B D8       501  ACCENT   LD HL,(OADR)
D329 1A             502           LD A,(DE)
D32A FE 5E          503           CP "^^"
D32C 28 0B          504           JR Z,ACC2
D32E 06 C7          505           LD B,0C7H
D330 FE 3D          506           CP "="
D332 28 12          507           JR Z,ACCSET
D334 36 80          508           LD (HL),80H
D336 C3 AA D1       509           JP UP0
D339 06 C8          510  ACC2     LD B,0C8H
D33B 13             511           INC DE
D33C 1A             512           LD A,(DE)
D33D FE 3D          513           CP "="
D33F 28 05          514           JR Z,ACCSET
D341 36 81          515           LD (HL),81H
D343 C3 AA D1       516           JP UP0
D346                517
D346 13             518  ACCSET   INC DE
D347 C3 0F D7       519           JP NORM2
D34A                520
D34A CD 68 D4       521  WAIT     CALL LKAZU
D34D D5             522           PUSH DE
D34E EB             523           EX DE,HL
D34F 2A 9B D8       524           LD HL,(OADR)
D352 7A             525           LD A,D
D353 B7             526           OR A
D354 28 06          527           JR Z,WAITNE1
D356 36 E1          528  WAITLP   LD (HL),0E1H
D358 23             529           INC HL
D359 3D             530           DEC A
D35A 20 FA          531           JR NZ,WAITLP
D35C 7B             532  WAITNE1 LD A,E
D35D B7             533           OR A
D35E 28 04          534           JR Z,WAITNE2
D360 36 E0          535           LD (HL),0E0H
D362 23             536           INC HL
D363 73             537           LD (HL),E
D364 D1             538  WAITNE2 POP DE
D365 C3 AA D1       539           JP UP0
D368                540
D368 2A 9B D8       541  BAR      LD HL,(OADR)
D36B 36 FF          542           LD (HL),0FFH
D36D C3 AA D1       543           JP UP0
D370                544
D370 2A 9B D8       545  REPIN    LD HL,(OADR)
D373 36 91          546           LD (HL),91H
D375 23             547           INC HL
D376 36 00          548           LD (HL),0
D378 C3 AA D1       549           JP UP0
D37B                550
D37B 2A 9B D8       551  REPOUT   LD HL,(OADR)
D37E 36 8C          552           LD (HL),8CH
D380 C3 AA D1       553           JP UP0
D383                554
D383 6F             555  OMPU     LD L,A
D384 26 00          556           LD H,0
D386 01 BB D3       557           LD BC,NNODT-41H
D389 09             558           ADD HL,BC
D38A 7E             559           LD A,(HL)
D38B F5             560           PUSH AF
D38C D5             561           PUSH DE
D38D 3A 99 D8       562           LD A,(OBUFF)
D390 6F             563           LD L,A
D391 26 00          564           LD H,0
D393 11 0C 00       565           LD DE,12
D396 CD 39 01       566           CALL MUL
D399 D1             567           POP DE
D39A F1             568           POP AF
D39B CD 55 DC       569           CALL SPSKIP
D39E 85             570           ADD A,L
D39F 47             571           LD B,A
D3A0 1A             572           LD A,(DE)
D3A1 FE 23          573           CP "#"
D3A3 20 04          574           JR NZ,OMPNE0
D3A5 04             575  OMPNEB0 INC B
D3A6 13             576           INC DE
D3A7 18 12          577           JR OMPNE1
D3A9 4F             578  OMPNE0   LD C,A
D3AA 3A BB D8       579           LD A,(KMFRAG)
D3AD B7             580           OR A
D3AE 20 0B          581           JR NZ,OMPNE1
D3B0 79             582           LD A,C
D3B1 FE 2B          583           CP "+"
D3B3 28 F0          584           JR Z,OMPNEB0
D3B5 FE 2D          585           CP "-"
D3B7 20 02          586           JR NZ,OMPNE1
D3B9 05             587           DEC B
D3BA 13             588           INC DE
D3BB 78             589  OMPNE1   LD A,B
D3BC FE 80          590           CP 80H
D3BE D2 24 D1       591           JP NC,ER3
D3C1 2A 9B D8       592           LD HL,(OADR)
D3C4 47             593           LD B,A
D3C5 3A BA D8       594           LD A,(TIEF)
D3C8 B7             595           OR A
D3C9 28 12          596           JR Z,OMPNEA
D3CB AF             597           XOR A
D3CC 32 BA D8       598           LD (TIEF),A
D3CF 3A B9 D8       599           LD A,(OMPF)
D3D2 B8             600           CP B
D3D3 28 08          601           JR Z,OMPNEA
D3D5 2B             602           DEC HL
D3D6 E5             603           PUSH HL
D3D7 2A B6 D8       604           LD HL,(BOADR)
D3DA 36 89          605           LD (HL),89H
D3DC E1             606           POP HL
D3DD 78             607  OMPNEA   LD A,B
D3DE 22 B6 D8       608           LD (BOADR),HL
D3E1 77             609           LD (HL),A
D3E2 F6 80          610           OR 80H
D3E4 32 B9 D8       611           LD (OMPF),A
D3E7 23             612  OMPUL    INC HL
D3E8 CD C4 D4       613           CALL LKAZU2
D3EB 47             614           LD B,A
D3EC 77             615           LD (HL),A
D3ED 3A B8 D8       616  OMPUL2   LD A,(MODE)
D3F0 B7             617           OR A
D3F1 CA AA D1       618           JP Z,UP0
D3F4 23             619           INC HL
D3F5 36 E0          620           LD (HL),0E0H
D3F7 23             621           INC HL
D3F8 70             622           LD (HL),B
D3F9 C3 AA D1       623           JP UP0
D3FC                624
D3FC 09 0B 00 02    625  NNODT    DB 9,11,0,2,4,5,7
D400 04 05 07
D403                626
D403 C5             627  KAZU     PUSH BC
D404 F5             628           PUSH AF
D405 CD 55 DC       629           CALL SPSKIP
D408 EB             630           EX DE,HL
D409 11 A3 D8       631           LD DE,SBUFF
D40C 06 08          632           LD B,8
D40E 7E             633  KAZULP   LD A,(HL)
D40F FE 30          634           CP 30H
D411 38 0C          635           JR C,KAZUNE1
D413 FE 3A          636           CP 3AH
D415 30 08          637           JR NC,KAZUNE1
D417 12             638           LD (DE),A
D418 23             639           INC HL
D419 13             640           INC DE
D41A 10 F2          641           DJNZ KAZULP
D41C C3 24 D1       642           JP ER3
D41F AF             643  KAZUNE1 XOR A
D420 12             644           LD (DE),A
D421 E5             645           PUSH HL
D422 21 A3 D8       646           LD HL,SBUFF
D425 7E             647           LD A,(HL)
D426 B7             648           OR A
D427 CA 24 D1       649           JP Z,ER3
D42A DF             650           RST 18H
D42B 13             651           DB 13H
D42C DA 24 D1       652           JP C,ER3
D42F E1             653           POP HL
D430 EB             654           EX DE,HL
D431 F1             655           POP AF
D432 C1             656           POP BC
D433 CD 55 DC       657           CALL SPSKIP
D436 C9             658           RET
D437                659
D437 E5             660  DKAZU    PUSH HL
D438 CD 03 D4       661           CALL KAZU
D43B 7C             662           LD A,H
D43C B7             663           OR A
D43D C2 24 D1       664           JP NZ,ER3
D440 7D             665           LD A,L
D441 FE 80          666           CP 80H
D443 D2 24 D1       667           JP NC,ER3
D446 E1             668           POP HL
D447 C9             669           RET
D448                670
D448 C5             671  HKAZU    PUSH BC
D449 F5             672           PUSH AF
D44A CD 55 DC       673           CALL SPSKIP
D44D EB             674           EX DE,HL
D44E 11 A3 D8       675           LD DE,SBUFF
D451 06 08          676           LD B,8
D453 7E             677  HKAZULP LD A,(HL)
D454 FE 2C          678           CP ","
D456 28 C7          679           JR Z,KAZUNE1
D458 FE 5D          680           CP "]"
D45A 28 C3          681           JR Z,KAZUNE1
D45C B7             682           OR A
D45D CA 24 D1       683           JP Z,ER3
```

▶DōGAのK氏にひと言いいたい。マルボロのカラーリングは、全然違うんだ！　全然違うんだったら。
新井　修一（18）千葉県

```
D460 12          684           LD (DE),A
D461 23          685           INC HL
D462 13          686           INC DE
D463 10 EE       687           DJNZ HKAZULP
D465 C3 24 D1    688           JP ER3
D468             689
D468 F5          690  LKAZU    PUSH AF
D469 CD 55 DC    691           CALL SPSKIP
D46C 1A          692           LD A,(DE)
D46D FE 21       693           CP "!"
D46F 28 48       694           JR Z,LDIR
D471 FE 30       695           CP 30H
D473 38 3E       696           JR C,LKAZUNE
D475 FE 3A       697           CP 3AH
D477 30 3A       698           JR NC,LKAZUNE
D479 CD 03 D4    699           CALL KAZU
D47C 7C          700           LD A,H
D47D B7          701           OR A
D47E C2 24 D1    702           JP NZ,ER3
D481 7D          703           LD A,L
D482 B7          704           OR A
D483 CA 24 D1    705           JP Z,ER3
D486 D5          706           PUSH DE
D487 EB          707           EX DE,HL
D488 21 C0 00    708           LD HL,192
D48B CD 53 01    709           CALL DIV
D48E D1          710           POP DE
D48F C5          711  LKANEX   PUSH BC
D490 7D          712           LD A,L
D491 47          713           LD B,A
D492 F5          714  LKALOOP  PUSH AF
D493 1A          715           LD A,(DE)
D494 FE 2E       716           CP "."
D496 20 0D       717           JR NZ,LKABRK
D498 F1          718           POP AF
D499 CB 38       719           SRL B
D49B 80          720           ADD A,B
D49C DA 24 D1    721           JP C,ER3
D49F 13          722           INC DE
D4A0 CD 55 DC    723           CALL SPSKIP
D4A3 18 ED       724           JR LKALOOP
D4A5 F1          725  LKABRK   POP AF
D4A6 C1          726           POP BC
D4A7 B7          727           OR A
D4A8 CA 24 D1    728           JP Z,ER3
D4AB 6F          729           LD L,A
D4AC 26 00       730           LD H,0
D4AE F1          731           POP AF
D4AF CD 55 DC    732           CALL SPSKIP
D4B2 C9          733           RET
D4B3             734
D4B3 3A 9A D8    735  LKAZUNE  LD A,(LBUFF)
D4B6 6F          736           LD L,A
D4B7 18 D6       737           JR LKANEX
D4B9             738
D4B9 13          739  LDIR     INC DE
D4BA F1          740           POP AF
D4BB CD 03 D4    741           CALL KAZU
D4BE 7C          742           LD A,H
D4BF B5          743           OR L
D4C0 CA 24 D1    744           JP Z,ER3
D4C3 C9          745           RET
D4C4             746
D4C4 E5          747  LKAZU2   PUSH HL
D4C5 CD 68 D4    748           CALL LKAZU
D4C8 7C          749           LD A,H
D4C9 B7          750           OR A
D4CA C2 24 D1    751           JP NZ,ER3
D4CD 7D          752           LD A,L
D4CE E1          753           POP HL
D4CF C9          754           RET
D4D0             755
D4D0 F5          756  FKAZU    PUSH AF
D4D1 CD 55 DC    757           CALL SPSKIP
D4D4 1A          758           LD A,(DE)
D4D5 FE 2B       759           CP "+"
D4D7 28 06       760           JR Z,PLUS
D4D9 FE 2D       761           CP "-"
D4DB 28 0E       762           JR Z,MINUS
D4DD 18 01       763           JR PLUSNE
D4DF 13          764  PLUS     INC DE
D4E0 CD 03 D4    765  PLUSNE   CALL KAZU
D4E3 7D          766           LD A,L
D4E4 FE 80       767           CP 80H
D4E6 D2 24 D1    768           JP NC,ER3
D4E9 18 0D       769           JR FKCHECK
D4EB 13          770  MINUS    INC DE
D4EC CD 03 D4    771           CALL KAZU
D4EF 7D          772           LD A,L
D4F0 ED 44       773           NEG
D4F2 FE 80       774           CP 80H
D4F4 DA 24 D1    775           JP C,ER3
D4F7 6F          776           LD L,A
D4F8 7C          777  FKCHECK  LD A,H
D4F9 B7          778           OR A
D4FA C2 24 D1    779           JP NZ,ER3
D4FD F1          780           POP AF
D4FE C9          781           RET
D4FF             782
D4FF             783           ;***** 1234 -> SET BIT *****
D4FF C5          784  BBSUB    PUSH BC
D500 0E 00       785           LD C,0
D502 CD 55 DC    786  BBLOOP   CALL SPSKIP
D505 1A          787           LD A,(DE)
D506 FE 31       788           CP 31H
D508 38 14       789           JR C,BBEND
D50A FE 35       790           CP 35H
D50C 30 10       791           JR NC,BBEND
D50E D6 30       792           SUB 30H
D510 06 00       793           LD B,0
D512 37          794           SCF
D513 CB 10       795  BBCLP    RL B
D515 3D          796           DEC A
D516 20 FB       797           JR NZ,BBCLP
D518 79          798           LD A,C
D519 B0          799           OR B
D51A 4F          800           LD C,A
D51B 13          801           INC DE
D51C 18 E4       802           JR BBLOOP
D51E 79          803  BBEND    LD A,C
D51F CD 55 DC    804           CALL SPSKIP
D522 C1          805           POP BC
D523 C9          806           RET
D524             807
D524             808
D524 2A 9B D8    809  KAKKO    LD HL,(OADR)
D527 36 98       810           LD (HL),98H
D529 23          811           INC HL
D52A CD FF D4    812           CALL BBSUB
D52D B7          813           OR A
D52E CA 24 D1    814           JP Z,ER3
D531 77          815           LD (HL),A
D532 E6 01       816           AND 1
D534 CC 89 D8    817           CALL Z,PMRES
D537 1A          818           LD A,(DE)
D538 FE 29       819           CP ")"
D53A C2 24 D1    820           JP NZ,ER3
D53D 23          821           INC HL
D53E 36 00       822           LD (HL),0
D540 13          823           INC DE
D541 C3 AA D1    824           JP UP0
D544             825
D544 CD 03 D4    826  SETCH    CALL KAZU
D547 7C          827           LD A,H
D548 B7          828           OR A
D549 C2 24 D1    829           JP NZ,ER3
D54C 7D          830           LD A,L
D54D 3D          831           DEC A
D54E FE 10       832           CP 10H
D550 D2 24 D1    833           JP NC,ER3
D553 2A 9B D8    834           LD HL,(OADR)
D556 36 C9       835           LD (HL),0C9H
D558 23          836           INC HL
D559 77          837           LD (HL),A
D55A C3 AA D1    838           JP UP0
D55D             839
D55D 2A 9B D8    840  VOLUME   LD HL,(OADR)
D560 36 9A       841           LD (HL),9AH
D562 CD 37 D4    842           CALL DKAZU
D565 FE 10       843           CP 16
D567 D2 24 D1    844           JP NC,ER3
D56A 23          845           INC HL
D56B B7          846           OR A
D56C 28 0C       847           JR Z,VL0
D56E 47          848           LD B,A
D56F 3E 0F       849           LD A,15
D571 90          850           SUB B
D572 07          851           RLCA
D573 07          852           RLCA
D574 47          853           LD B,A
D575 3E 7F       854           LD A,127
D577 90          855           SUB B
D578 18 01       856           JR VO0NE
D57A AF          857  VL0      XOR A
D57B 77          858  VO0NE    LD (HL),A
D57C C3 AA D1    859           JP UP0
D57F             860
D57F 1A          861  EXP3     LD A,(DE)
D580 2A 9B D8    862           LD HL,(OADR)
D583 13          863           INC DE
D584 CD 1B D1    864           CALL UTOL
D587 FE 54       865           CP "T"
D589 28 0B       866           JR Z,?T
D58B FE 56       867           CP "V"
D58D 28 0B       868           JR Z,?V
D58F FE 42       869           CP "B"
D591 28 0B       870           JR Z,?B
D593 C3 24 D1    871           JP ER3
D596             872
D596 36 E9       873  ?T       LD (HL),0E9H
D598 18 06       874           JR ?SET
D59A 36 EA       875  ?V       LD (HL),0EAH
D59C 18 02       876           JR ?SET
D59E 36 EB       877  ?B       LD (HL),0EBH
D5A0             878
D5A0 23          879  ?SET     INC HL
D5A1 CD C4 D4    880           CALL LKAZU2
D5A4 77          881           LD (HL),A
D5A5 C3 AA D1    882           JP UP0
D5A8             883
D5A8 1A          884  EXP1     LD A,(DE)
D5A9 13          885           INC DE
D5AA CD 1B D1    886           CALL UTOL
D5AD 4F          887           LD C,A
D5AE 21 C6 D5    888           LD HL,E1COM
D5B1 7E          889  E1LOOP   LD A,(HL)
D5B2 B7          890           OR A
D5B3 CA EE D5    891           JP Z,NEIRO
D5B6 23          892           INC HL
D5B7 B9          893           CP C
D5B8 28 04       894           JR Z,E1FIND
D5BA 23          895           INC HL
D5BB 23          896           INC HL
D5BC 18 F3       897           JR E1LOOP
D5BE 4E          898  E1FIND   LD C,(HL)
D5BF 23          899           INC HL
D5C0 66          900           LD H,(HL)
D5C1 69          901           LD L,C
D5C2 CD 55 DC    902           CALL SPSKIP
D5C5 E9          903           JP (HL)
D5C6             904
D5C6 56          905  E1COM    DM "V"
D5C7 F3 D5       906           DW VOL
D5C9 50          907           DM "P"
D5CA F7 D5       908           DW PAN
D5CC 45          909           DM "E"
D5CD FB D5       910           DW EXP
D5CF 4D          911           DM "M"
D5D0 FF D5       912           DW MOD
D5D2 4B          913           DM "K"
D5D3 0D D6       914           DW KEY
D5D5 53          915           DM "S"
D5D6 24 D6       916           DW SHIFT
D5D8 3C          917           DM "<"
D5D9 32 D6       918           DW CRESC
D5DB 3E          919           DM ">"
D5DC 36 D6       920           DW DIM
D5DE 41          921           DM "A"
D5DF CE D6       922           DW ACCEL
D5E1 52          923           DM "R"
D5E2 D2 D6       924           DW RIT
D5E4 42          925           DM "B"
D5E5 F5 D6       926           DW BEND
D5E7 4E          927           DM "N"
D5E8 0D D7       928           DW NORM
D5EA 43          929           DM "C"
D5EB 21 D7       930           DW CHANGE
D5ED 00          931           DB 0
D5EE             932
D5EE 3E 99       933  NEIRO    LD A,99H
D5F0 1B          934           DEC DE
D5F1 18 0E       935           JR E1SET
D5F3 3E 9A       936  VOL      LD A,9AH
D5F5 18 0A       937           JR E1SET
D5F7 3E 9B       938  PAN      LD A,9BH
D5F9 18 06       939           JR E1SET
D5FB 3E 9C       940  EXP      LD A,9CH
D5FD 18 02       941           JR E1SET
D5FF 3E 9D       942  MOD      LD A,9DH
D601 2A 9B D8    943  E1SET    LD HL,(OADR)
D604 77          944           LD (HL),A
D605 CD 37 D4    945           CALL DKAZU
D608 23          946           INC HL
D609 77          947           LD (HL),A
D60A C3 AA D1    948           JP UP0
D60D             949
D60D CD D0 D4    950  KEY      CALL FKAZU
D610 7D          951           LD A,L
D611 FE F9       952           CP -7
D613 30 05       953           JR NC,KEYPS
D615 FE 08       954           CP 8
D617 D2 24 D1    955           JP NC,ER3
D61A 2A 9B D8    956  KEYPS    LD HL,(OADR)
D61D 36 E2       957           LD (HL),0E2H
D61F 23          958           INC HL
D620 77          959           LD (HL),A
D621 C3 AA D1    960           JP UP0
D624             961
```

▶最近“ざ・ほろすこーぷ”のハートマークが少ない（というかまったくないみたい)。ひょっとしてひょっとすると，これはみんなOh!Xのせいなんだろうか。

森　星児（17）島根県

```
D624 CD D0 D4     962  SHIFT   CALL FKAZU
D627 7D           963          LD A,L
D628 2A 9B D8     964          LD HL,(OADR)
D62B 36 E5        965          LD (HL),0E5H
D62D 23           966          INC HL
D62E 77           967          LD (HL),A
D62F C3 AA D1     968          JP UP0
D632              969
D632 3E B0        970  CRESC   LD A,0B0H
D634 18 02        971          JR DIM2
D636 3E B1        972  DIM     LD A,0B1H
D638 2A 9B D8     973  DIM2    LD HL,(OADR)
D63B 77           974          LD (HL),A
D63C CD B3 D6     975          CALL CURV
D63F 1A           976          LD A,(DE)
D640 FE 5B        977          CP "["
D642 20 56        978          JR NZ,DIMDK
D644 13           979          INC DE
D645 1A           980          LD A,(DE)
D646 13           981          INC DE
D647 CD 1B D1     982          CALL UTOL
D64A 4F           983          LD C,A
D64B 1A           984          LD A,(DE)
D64C CD 1B D1     985          CALL UTOL
D64F 47           986          LD B,A
D650 79           987          LD A,C
D651 FE 46        988          CP "F"
D653 28 0B        989          JR Z,DIMF
D655 FE 50        990          CP "P"
D657 28 15        991          JR Z,DIMP
D659 FE 4D        992          CP "M"
D65B 28 1F        993          JR Z,DIMM
D65D C3 24 D1     994          JP ER3
D660 78           995  DIMF    LD A,B
D661 FE 46        996          CP "F"
D663 28 04        997          JR Z,DIMFF
D665 3E 84        998          LD A,84H
D667 18 26        999          JR DIMSX
D669 3E 85       1000  DIMFF   LD A,85H
D66B 13          1001          INC DE
D66C 18 21       1002          JR DIMSX
D66E 78          1003  DIMP    LD A,B
D66F FE 50       1004          CP "P"
D671 28 04       1005          JR Z,DIMPP
D673 3E 81       1006          LD A,81H
D675 18 18       1007          JR DIMSX
D677 3E 80       1008  DIMPP   LD A,80H
D679 13          1009          INC DE
D67A 18 13       1010          JR DIMSX
D67C 78          1011  DIMM    LD A,B
D67D 13          1012          INC DE
D67E FE 46       1013          CP "F"
D680 28 07       1014          JR Z,DIMMF
D682 FE 50       1015          CP "P"
D684 28 07       1016          JR Z,DIMMP
D686 C3 24 D1    1017          JP ER3
D689 3E 83       1018  DIMMF   LD A,83H
D68B 18 02       1019          JR DIMSX
D68D 3E 82       1020  DIMMP   LD A,82H
D68F 47          1021  DIMSX   LD B,A
D690 1A          1022          LD A,(DE)
D691 FE 5D       1023          CP "]"
D693 C2 24 D1    1024          JP NZ,ER3
D696 13          1025          INC DE
D697 78          1026          LD A,B
D698 18 03       1027          JR DIMST
D69A CD 37 D4    1028  DIMDK   CALL DKAZU
D69D 77          1029  DIMST   LD (HL),A
D69E 1A          1030          LD A,(DE)
D69F FE 2C       1031          CP ","
D6A1 C2 24 D1    1032          JP NZ,ER3
D6A4 13          1033  DIMX    INC DE
D6A5 23          1034          INC HL
D6A6 E5          1035          PUSH HL
D6A7 CD 68 D4    1036          CALL LKAZU
D6AA 4D          1037          LD C,L
D6AB 44          1038          LD B,H
D6AC E1          1039          POP HL
D6AD 71          1040          LD (HL),C
D6AE 23          1041          INC HL
D6AF 70          1042          LD (HL),B
D6B0 C3 AA D1    1043          JP UP0
D6B3             1044
D6B3 23          1045  CURV    INC HL
D6B4 E5          1046          PUSH HL
D6B5 CD 03 D4    1047          CALL KAZU
D6B8 7C          1048          LD A,H
D6B9 B7          1049          OR A
D6BA C2 24 D1    1050          JP NZ,ER3
D6BD 7D          1051          LD A,L
D6BE FE 20       1052          CP 32
D6C0 D2 24 D1    1053          JP NC,ER3
D6C3 E1          1054          POP HL
D6C4 77          1055          LD (HL),A
D6C5 23          1056          INC HL
D6C6 1A          1057          LD A,(DE)
D6C7 FE 2C       1058          CP ","
D6C9 C2 24 D1    1059          JP NZ,ER3
D6CC 13          1060          INC DE
D6CD C9          1061          RET
D6CE             1062
D6CE 3E B2       1063  ACCEL   LD A,0B2H
D6D0 18 02       1064          JR RIT2
D6D2 3E B3       1065  RIT     LD A,0B3H
D6D4 2A 9B D8    1066  RIT2    LD HL,(OADR)
D6D7 77          1067          LD (HL),A
D6D8 CD B3 D6    1068          CALL CURV
D6DB E5          1069          PUSH HL
D6DC CD 03 D4    1070          CALL KAZU
D6DF 7C          1071          LD A,H
D6E0 B7          1072          OR A
D6E1 C2 24 D1    1073          JP NZ,ER3
D6E4 7D          1074          LD A,L
D6E5 FE 1E       1075          CP 30
D6E7 DA 24 D1    1076          JP C,ER3
D6EA E1          1077          POP HL
D6EB 77          1078          LD (HL),A
D6EC 1A          1079          LD A,(DE)
D6ED FE 2C       1080          CP ","
D6EF C2 24 D1    1081          JP NZ,ER3
D6F2 C3 A4 D6    1082          JP DIMX
D6F5             1083
D6F5 2A 9B D8    1084  BEND    LD HL,(OADR)
D6F8 36 B4       1085          LD (HL),0B4H
D6FA CD B3 D6    1086          CALL CURV
D6FD E5          1087          PUSH HL
D6FE CD D0 D4    1088          CALL FKAZU
D701 7D          1089          LD A,L
D702 E1          1090          POP HL
D703 77          1091          LD (HL),A
D704 1A          1092          LD A,(DE)
D705 FE 2C       1093          CP ","
D707 C2 24 D1    1094          JP NZ,ER3
D70A C3 A4 D6    1095          JP DIMX
D70D             1096
D70D 06 C6       1097  NORM    LD B,0C6H
D70F C5          1098  NORM2   PUSH BC
D710 CD 37 D4    1099          CALL DKAZU
D713 B7          1100          OR A
D714 CA 24 D1    1101          JP Z,ER3
D717 2A 9B D8    1102          LD HL,(OADR)
D71A C1          1103          POP BC
D71B 70          1104          LD (HL),B
D71C 23          1105          INC HL
D71D 77          1106          LD (HL),A
D71E C3 AA D1    1107          JP UP0
D721             1108
D721 2A 9B D8    1109  CHANGE  LD HL,(OADR)
D724 36 E7       1110          LD (HL),0E7H
D726 CD 37 D4    1111          CALL DKAZU
D729 23          1112          INC HL
D72A 77          1113          LD (HL),A
D72B 1A          1114          LD A,(DE)
D72C FE 2C       1115          CP ","
D72E C2 24 D1    1116          JP NZ,ER3
D731 13          1117          INC DE
D732 CD 37 D4    1118          CALL DKAZU
D735 23          1119          INC HL
D736 77          1120          LD (HL),A
D737 C3 AA D1    1121          JP UP0
D73A             1122
D73A 21 64 D7    1123  EXP2    LD HL,E2COM
D73D D5          1124  E2LOOP  PUSH DE
D73E 7E          1125          LD A,(HL)
D73F B7          1126          OR A
D740 CA 24 D1    1127          JP Z,ER3
D743 7E          1128  E2LOOP2 LD A,(HL)
D744 B7          1129          OR A
D745 28 16       1130          JR Z,E2FIND
D747 47          1131          LD B,A
D748 1A          1132          LD A,(DE)
D749 CD 1B D1    1133          CALL UTOL
D74C B8          1134          CP B
D74D 20 04       1135          JR NZ,E2SKIP
D74F 23          1136          INC HL
D750 13          1137          INC DE
D751 18 F0       1138          JR E2LOOP2
D753 7E          1139  E2SKIP  LD A,(HL)
D754 23          1140          INC HL
D755 B7          1141          OR A
D756 20 FB       1142          JR NZ,E2SKIP
D758 23          1143          INC HL
D759 23          1144          INC HL
D75A D1          1145          POP DE
D75B 18 E0       1146          JR E2LOOP
D75D C1          1147  E2FIND  POP BC
D75E 23          1148          INC HL
D75F 4E          1149          LD C,(HL)
D760 23          1150          INC HL
D761 66          1151          LD H,(HL)
D762 69          1152          LD L,C
D763 E9          1153          JP (HL)
D764             1154
D764 24          1155  E2COM   DM "$"
D765 00          1156          DB 0
D766 C5 D7       1157          DW SENYO
D768 44 43       1158          DM "DC"
D76A 00          1159          DB 0
D76B C9 D7       1160          DW DC
D76D 44 53       1161          DM "DS"
D76F 00          1162          DB 0
D770 D0 D7       1163          DW DS
D772 54 4F       1164          DM "TO"
D774 00          1165          DB 0
D775 D7 D7       1166          DW TO
D777 43 4F 44 41 1167          DM "CODA"
D77B 00          1168          DB 0
D77C DB D7       1169          DW CODA
D77E 46 49 4E 45 1170          DM "FINE"
D782 00          1171          DB 0
D783 E2 D7       1172          DW FINE
D785 46 4F       1173          DM "FO"
D787 00          1174          DB 0
D788 F7 D7       1175          DW FO
D78A 50 4F 4C 59 1176          DM "POLY"
D78E 00          1177          DB 0
D78F 6E D8       1178          DW POLY
D791 4D 4F 4E 4F 1179          DM "MONO"
D795 00          1180          DB 0
D796 7B D8       1181          DW MONO
D798 53 4D 4F 44 1182          DM "SMODE"
D79C 45
D79D 00          1183          DB 0
D79E 0F D8       1184          DW SMODE
D7A0 4B 4D 4F 44 1185          DM "KMODE"
D7A4 45
D7A5 00          1186          DB 0
D7A6 24 D8       1187          DW KMODE
D7A8 50 50       1188          DM "PP"
D7AA 00          1189          DB 0
D7AB 35 D8       1190          DW PP
D7AD 50          1191          DM "P"
D7AE 00          1192          DB 0
D7AF 3B D8       1193          DW P
D7B1 4D 50       1194          DM "MP"
D7B3 00          1195          DB 0
D7B4 41 D8       1196          DW MP
D7B6 4D 46       1197          DM "MF"
D7B8 00          1198          DB 0
D7B9 47 D8       1199          DW MF
D7BB 46 46       1200          DM "FF"
D7BD 00          1201          DB 0
D7BE 53 D8       1202          DW FF
D7C0 46          1203          DM "F"
D7C1 00          1204          DB 0
D7C2 4D D8       1205          DW F
D7C4 00          1206          DB 0
D7C5             1207
D7C5 3E 94       1208  SENYO   LD A,94H
D7C7 18 1B       1209          JR LPSET
D7C9 CD 89 D8    1210  DC      CALL PMRES
D7CC 3E 92       1211          LD A,92H
D7CE 18 14       1212          JR LPSET
D7D0 CD 89 D8    1213  DS      CALL PMRES
D7D3 3E 93       1214          LD A,93H
D7D5 18 0D       1215          JR LPSET
D7D7 3E 95       1216  TO      LD A,95H
D7D9 18 09       1217          JR LPSET
D7DB CD 89 D8    1218  CODA    CALL PMRES
D7DE 3E 96       1219          LD A,96H
D7E0 18 02       1220          JR LPSET
D7E2 3E 97       1221  FINE    LD A,97H
D7E4 2A 9B D8    1222  LPSET   LD HL,(OADR)
D7E7 77          1223          LD (HL),A
D7E8 23          1224          INC HL
D7E9 CD FF D4    1225          CALL BBSUB
D7EC B7          1226          OR A
D7ED 20 01       1227          JR NZ,LPSNE1
D7EF 3C          1228          INC A
D7F0 77          1229  LPSNE1  LD (HL),A
D7F1 CD 91 D8    1230          CALL CEK]
D7F4 C3 AA D1    1231          JP UP0
D7F7             1232
D7F7             1233
D7F7 2A 9B D8    1234  FO      LD HL,(OADR)
D7FA 36 A0       1235          LD (HL),0A0H
D7FC CD B3 D6    1236          CALL CURV
D7FF E5          1237          PUSH HL
```

▶なんと，あのMZ-1500用DARK STORMの音楽は古代祐三氏の作品なんだよー。えっ，知ってるって？ CD出ないかな。　小口　卓（16）千葉県

```
D800 CD 68 D4     1238            CALL LKAZU
D803 4D           1239            LD C,L
D804 44           1240            LD B,H
D805 E1           1241            POP HL
D806 71           1242            LD (HL),C
D807 23           1243            INC HL
D808 70           1244            LD (HL),B
D809 CD 91 D8     1245            CALL CEK]
D80C C3 AA D1     1246            JP UP0
D80F              1247
D80F CD 37 D4     1248    SMODE   CALL DKAZU
D812 FE 04        1249            CP 4
D814 D2 24 D1     1250            JP NC,ER3
D817 2A 9B D8     1251            LD HL,(OADR)
D81A 36 E8        1252            LD (HL),0E8H
D81C 23           1253            INC HL
D81D 77           1254            LD (HL),A
D81E CD 91 D8     1255            CALL CEK]
D821 C3 AA D1     1256            JP UP0
D824              1257
D824 CD 37 D4     1258    KMODE   CALL DKAZU
D827 FE 02        1259            CP 2
D829 D2 24 D1     1260            JP NC,ER3
D82C 32 BB D8     1261            LD (KMFRAG),A
D82F CD 91 D8     1262            CALL CEK]
D832 C3 AE D1     1263            JP UPX0
D835              1264
D835 3E 82        1265    PP      LD A,82H
D837 06 C0        1266            LD B,0C0H
D839 18 1C        1267            JR KKSET
D83B 3E 83        1268    P       LD A,83H
D83D 06 C1        1269            LD B,0C1H
D83F 18 16        1270            JR KKSET
D841 3E 84        1271    MP      LD A,84H
D843 06 C2        1272            LD B,0C2H
D845 18 10        1273            JR KKSET
D847 3E 85        1274    MF      LD A,85H
D849 06 C3        1275            LD B,0C3H
D84B 18 0A        1276            JR KKSET
D84D 3E 86        1277    F       LD A,86H
D84F 06 C4        1278            LD B,0C4H
D851 18 04        1279            JR KKSET
D853 3E 87        1280    FF      LD A,87H
D855 06 C5        1281            LD B,0C5H
D857 2A 9B D8     1282    KKSET   LD HL,(OADR)
D85A 77           1283            LD (HL),A
D85B CD 91 D8     1284            CALL CEK]
D85E 1A           1285            LD A,(DE)
D85F FE 3D        1286            CP "="
D861 C2 AA D1     1287            JP NZ,UP0
D864 70           1288            LD (HL),B
D865 13           1289            INC DE
D866 23           1290            INC HL
D867 CD 37 D4     1291            CALL DKAZU
D86A 77           1292            LD (HL),A
D86B C3 AA D1     1293            JP UP0
D86E              1294
D86E AF           1295    POLY    XOR A
D86F 32 B8 D8     1296            LD (MODE),A
D872 CD 89 D8     1297            CALL PMRES
D875 CD 91 D8     1298            CALL CEK]
D878 C3 AE D1     1299            JP UPX0
D87B              1300
D87B 3E 80        1301    MONO    LD A,80H
D87D 32 B8 D8     1302            LD (MODE),A
D880 CD 89 D8     1303            CALL PMRES
D883 CD 91 D8     1304            CALL CEK]
D886 C3 AE D1     1305            JP UPX0
D889              1306
D889 AF           1307    PMRES   XOR A
D88A 32 B9 D8     1308            LD (OMPF),A
D88D 32 BA D8     1309            LD (TIEF),A
D890 C9           1310            RET
D891              1311
D891 1A           1312    CEK]    LD A,(DE)
D892 FE 5D        1313            CP "]"
D894 C2 24 D1     1314            JP NZ,ER3
D897 13           1315            INC DE
D898 C9           1316            RET
D899              1317
D899 05           1318    OBUFF   DB 5
D89A 30           1319    LBUFF   DB 48
D89B 00 40        1320    OADR    DW 4000H
D89D              1321    SADR    DS 2
D89F              1322    SEADR   DS 2
D8A1              1323    SPBUFF  DS 2
D8A3              1324    SBUFF   DS 10
D8AD              1325    MABUFF  DS 8
D8B5 3D           1326    NEWF    DB 3DH
D8B6              1327    BOADR   DS 2
D8B8 00           1328    MODE    DB 0
D8B9 00           1329    OMPF    DB 0
D8BA 00           1330    TIEF    DB 0
D8BB 01           1331    KMFRAG  DB 1
D8BC              1332
D8BC 3E 2B        1333    ER43    LD A,43
D8BE C3 7F D0     1334            JP ERROR
D8C1 3E 06        1335    ER6X    LD A,6
D8C3 C3 7F D0     1336            JP ERROR
D8C6              1337
D8C6              1338            ;***** TOTAL SYSTEM COTROLER *****
D8C6 F5           1339    OPEN    PUSH AF
D8C7 78           1340            LD A,B
D8C8 FE 02        1341            CP 2
D8CA C2 47 DA     1342            JP NZ,ER4X                 ;Type mismatch error
D8CD 3A 30 FF     1343            LD A,(OPENF)
D8D0 FE 55        1344            CP 55H
D8D2 28 E8        1345            JR Z,ER43                  ;Already open error
D8D4 E5           1346            PUSH HL
D8D5 D5           1347            PUSH DE
D8D6 C5           1348            PUSH BC
D8D7 AF           1349            XOR A
D8D8 32 B8 D8     1350            LD (MODE),A
D8DB 32 BA D8     1351            LD (TIEF),A
D8DE 32 B9 D8     1352            LD (OMPF),A
D8E1 3C           1353            INC A
D8E2 32 BB D8     1354            LD (KMFRAG),A
D8E5 06 10        1355            LD B,TRK
D8E7 21 52 FF     1356            LD HL,TRKMAP
D8EA 36 FF        1357    OPCLLP  LD (HL),0FFH
D8EC 23           1358            INC HL
D8ED 10 FB        1359            DJNZ OPCLLP
D8EF 3E 55        1360            LD A,55H
D8F1 32 30 FF     1361            LD (OPENF),A    ;FRAG
D8F4 AF           1362            XOR A
D8F5 32 93 FF     1363            LD (SHIFT1),A   ;KEY SHIFT RESET
D8F8 32 92 FF     1364            LD (TRKNO),A    ;TRACK
D8FB CD B9 D0     1365            CALL NEWSEA
D8FE 38 C1        1366            JR C,ER6X
D900 32 32 FF     1367            LD (USEMAP),A
D903 AF           1368            XOR A
D904 32 52 FF     1369            LD (TRKMAP),A
D907 3C           1370            INC A
D908 32 31 FF     1371            LD (USEMAPC),A
D90B 21 30 20     1372            LD HL,DTSTART
D90E 22 62 FF     1373            LD (TRKADR),HL
D911 22 9B D8     1374            LD (OADR),HL
D914 3E 3D        1375            LD A,3DH
D916 32 B5 D8     1376            LD (NEWF),A
```

```
D919 3E 05        1377            LD A,5
D91B 32 99 D8     1378            LD (OBUFF),A
D91E 3E 30        1379            LD A,48
D920 32 9A D8     1380            LD (LBUFF),A
D923 C1           1381            POP BC
D924 D1           1382            POP DE
D925 E1           1383            POP HL
D926 06 02        1384            LD B,2
D928 36 01        1385            LD (HL),1
D92A 23           1386            INC HL
D92B 36 00        1387            LD (HL),0
D92D 2B           1388            DEC HL
D92E F1           1389            POP AF
D92F CD C0 D9     1390            CALL ENDMARK
D932 C9           1391            RET
D933              1392
D933 3E 03        1393    ER3X    LD A,3
D935 C3 7F D0     1394            JP ERROR
D938              1395    NEWTRACK
D938 F5           1396            PUSH AF
D939 78           1397            LD A,B
D93A FE 02        1398            CP 2
D93C C2 47 DA     1399            JP NZ,ER4X                 ;Type mismatch error
D93F 3A 30 FF     1400            LD A,(OPENF)
D942 FE 55        1401            CP 55H
D944 C2 7D D0     1402            JP NZ,ER44                 ;Not open error
D947 3A 92 FF     1403            LD A,(TRKNO)
D94A FE 0F        1404            CP 15
D94C 30 E5        1405            JR NC,ER3X
D94E E5           1406            PUSH HL
D94F D5           1407            PUSH DE
D950 C5           1408            PUSH BC
D951 AF           1409            XOR A
D952 32 B8 D8     1410            LD (MODE),A
D955 32 B9 D8     1411            LD (OMPF),A
D958 32 BA D8     1412            LD (TIEF),A
D95B 3C           1413            INC A
D95C 32 BB D8     1414            LD (KMFRAG),A
D95F 3E 05        1415            LD A,5
D961 32 99 D8     1416            LD (OBUFF),A
D964 3E 30        1417            LD A,48
D966 32 9A D8     1418            LD (LBUFF),A
D969 3A 92 FF     1419            LD A,(TRKNO)
D96C 3C           1420            INC A
D96D 32 92 FF     1421            LD (TRKNO),A
D970 4F           1422            LD C,A
D971 06 00        1423            LD B,0
D973 21 51 FF     1424            LD HL,TRKMAP-1
D976 09           1425            ADD HL,BC
D977 ED 5B 9B D8  1426            LD DE,(OADR)
D97B 13           1427            INC DE
D97C 7A           1428            LD A,D
D97D E6 E0        1429            AND 0E0H
D97F 07           1430            RLCA
D980 07           1431            RLCA
D981 07           1432            RLCA
D982 3D           1433            DEC A
D983 86           1434            ADD A,(HL)
D984 23           1435            INC HL
D985 77           1436            LD (HL),A
D986 21 31 FF     1437            LD HL,USEMAPC
D989 3C           1438            INC A
D98A BE           1439            CP (HL)
D98B F5           1440            PUSH AF
D98C 21 62 FF     1441            LD HL,TRKADR
D98F 09           1442            ADD HL,BC
D990 09           1443            ADD HL,BC
D991 ED 5B 9B D8  1444            LD DE,(OADR)
D995 13           1445            INC DE
D996 7A           1446            LD A,D
D997 E6 1F        1447            AND 1FH
D999 F6 20        1448            OR 20H
D99B 57           1449            LD D,A
D99C ED 53 9B D8  1450            LD (OADR),DE
D9A0 73           1451            LD (HL),E
D9A1 23           1452            INC HL
D9A2 72           1453            LD (HL),D
D9A3 F1           1454            POP AF
D9A4 20 04        1455            JR NZ,NFS5D
D9A6 3E 3D        1456            LD A,3DH
D9A8 18 02        1457            JR NFSNE
D9AA 3E 5D        1458    NFS5D   LD A,5DH
D9AC 32 B5 D8     1459    NFSNE   LD (NEWF),A
D9AF 79           1460            LD A,C
D9B0 3C           1461            INC A
D9B1 C1           1462            POP BC
D9B2 D1           1463            POP DE
D9B3 E1           1464            POP HL
D9B4 06 02        1465            LD B,2
D9B6 77           1466            LD (HL),A
D9B7 23           1467            INC HL
D9B8 36 00        1468            LD (HL),0
D9BA 2B           1469            DEC HL
D9BB F1           1470            POP AF
D9BC CD C0 D9     1471            CALL ENDMARK
D9BF C9           1472            RET
D9C0              1473
D9C0 F5           1474    ENDMARK PUSH AF
D9C1 E5           1475            PUSH HL
D9C2 D5           1476            PUSH DE
D9C3 C5           1477            PUSH BC
D9C4 3A 92 FF     1478            LD A,(TRKNO)
D9C7 57           1479            LD D,A
D9C8 4F           1480            LD C,A
D9C9 06 00        1481            LD B,0
D9CB 21 52 FF     1482            LD HL,TRKMAP
D9CE 09           1483            ADD HL,BC
D9CF 4E           1484            LD C,(HL)
D9D0 21 32 FF     1485            LD HL,USEMAP
D9D3 09           1486            ADD HL,BC
D9D4 F3           1487            DI                  ;SET MAPPING
D9D5 3E 01        1488            LD A,1
D9D7 D3 B4        1489            OUT (0B4H),A
D9D9 DB B5        1490            IN A,(0B5H)
D9DB F5           1491            PUSH AF
D9DC 3E 01        1492            LD A,1
D9DE D3 B4        1493            OUT (0B4H),A
D9E0 7E           1494            LD A,(HL)
D9E1 D3 B5        1495            OUT (0B5H),A
D9E3 FB           1496            EI
D9E4 4A           1497            LD C,D
D9E5 21 62 FF     1498            LD HL,TRKADR
D9E8 09           1499            ADD HL,BC
D9E9 09           1500            ADD HL,BC
D9EA 5E           1501            LD E,(HL)
D9EB 23           1502            INC HL
D9EC 56           1503            LD D,(HL)
D9ED 3E FE        1504            LD A,0FEH           ;END MARK
D9EF 12           1505            LD (DE),A
D9F0 F3           1506            DI                  ;RESET MAPPING
D9F1 3E 01        1507            LD A,1
D9F3 D3 B4        1508            OUT (0B4H),A
D9F5 F1           1509            POP AF
D9F6 D3 B5        1510            OUT (0B5H),A
D9F8 FB           1511            EI
D9F9 C1           1512            POP BC
D9FA D1           1513            POP DE
D9FB E1           1514            POP HL
D9FC F1           1515            POP AF
```


▶八王子市のＪ＆Ｐでは，FM TOWNSのアフターバーナーとX68000のアフターバーナーを並べてデモっています。ムシャクシャしているときに見比べるとスカットしますよ。Ｊ＆Ｐもいいことしてくれるなぁ。　　高橋　悟（18）東京都

```
D9FD C9             1516            RET
D9FE                1517
D9FE F5             1518  CLOSE     PUSH AF
D9FF 78             1519            LD A,B
DA00 FE 02          1520            CP 2
DA02 C2 47 DA       1521            JP NZ,ER4X                  ;Type mismatch error
DA05 3A 30 FF       1522            LD A,(OPENF)
DA08 FE 55          1523            CP 55H
DA0A 28 07          1524            JR Z,CLOSES
DA0C AF             1525            XOR A                       ;Not open
DA0D 77             1526            LD (HL),A
DA0E 23             1527            INC HL
DA0F 77             1528            LD (HL),A
DA10 2B             1529            DEC HL
DA11 F1             1530            POP AF
DA12 C9             1531            RET
DA13 E5             1532  CLOSES    PUSH HL
DA14 D5             1533            PUSH DE
DA15 C5             1534            PUSH BC
DA16 CD 26 DA       1535            CALL CLOSEX
DA19 C1             1536            POP BC
DA1A D1             1537            POP DE
DA1B E1             1538            POP HL
DA1C 06 02          1539            LD B,2
DA1E 36 01          1540            LD (HL),1
DA20 23             1541            INC HL
DA21 36 00          1542            LD (HL),0
DA23 2B             1543            DEC HL
DA24 F1             1544            POP AF
DA25 C9             1545            RET
DA26                1546
DA26 CD 28 DC       1547  CLOSEX    CALL STOPX
DA29 AF             1548            XOR A
DA2A 32 30 FF       1549            LD (OPENF),A
DA2D 3A 31 FF       1550            LD A,(USEMAPC)
DA30 B7             1551            OR A
DA31 C8             1552            RET Z
DA32 11 32 FF       1553            LD DE,USEMAP
DA35 F5             1554  CLOOP     PUSH AF
DA36 1A             1555            LD A,(DE)
DA37 4F             1556            LD C,A
DA38 06 00          1557            LD B,0
DA3A 21 40 05       1558            LD HL,540H
DA3D 09             1559            ADD HL,BC
DA3E CB BE          1560            RES 7,(HL)
DA40 13             1561            INC DE
DA41 3D             1562            DEC A
DA42 F1             1563            POP AF
DA43 3D             1564            DEC A
DA44 20 EF          1565            JR NZ,CLOOP
DA46 C9             1566            RET
DA47                1567
DA47 3E 04          1568  ER4X      LD A,4
DA49 C3 7F D0       1569            JP ERROR
DA4C                1570
DA4C F5             1571  PLAY      PUSH AF
DA4D 78             1572            LD A,B
DA4E FE 02          1573            CP 2
DA50 20 F5          1574            JR NZ,ER4X                  ;Type mismatch error
DA52 3A 30 FF       1575            LD A,(OPENF)
DA55 FE 55          1576            CP 55H
DA57 C2 7D D0       1577            JP NZ,ER44                  ;Not open error
DA5A C5             1578            PUSH BC
DA5B D5             1579            PUSH DE
DA5C FD E5          1580            PUSH IY
DA5E E5             1581            PUSH HL
DA5F 3E 03          1582            LD A,3
DA61 D3 47          1583            OUT (CTC3),A
DA63 FB             1584            EI
DA64 3A 81 09       1585  PLAYW     LD A,(#DC1)                 ;WAIT UNTIL ALL DATA OUT
PUT
DA67 B7             1586            OR A
DA68 20 FA          1587            JR NZ,PLAYW
DA6A 21 F7 DB       1588            LD HL,SIOCD
DA6D 01 43 0A       1589            LD BC,0A00H+SIOC
DA70 ED B3          1590            OTIR
DA72 21 00 00       1591            LD HL,0                     ;INIT MIDI OUT INTSUB
DA75 22 82 09       1592            LD (#DC1+1),HL
DA78 22 6C E1       1593            LD (#DC2),HL
DA7B 22 6E E1       1594            LD (#DC2+2),HL
DA7E CD 28 DC       1595            CALL STOPX
DA81 21 00 EE       1596            LD HL,TRKWORK
DA84 11 01 EE       1597            LD DE,TRKWORK+1
DA87 01 2F 11       1598            LD BC,TRK*TLEN+MIDIX+SYSLEN-1
DA8A 36 00          1599            LD (HL),0
DA8C ED B0          1600            LDIR
DA8E 21 A0 FF       1601            LD HL,RTBUFF
DA91 11 A1 FF       1602            LD DE,RTBUFF+1
DA94 01 0F 00       1603            LD BC,0FH
DA97 36 00          1604            LD (HL),0
DA99 ED B0          1605            LDIR
DA9B 21 AB FF       1606            LD HL,RTBUFF+0BH
DA9E 3E 01          1607            LD A,1
DAA0 77             1608            LD (HL),A
DAA1 23             1609            INC HL
DAA2 77             1610            LD (HL),A
DAA3 FD 21 00 EE    1611            LD IY,TRKWORK
DAA7 11 62 FF       1612            LD DE,TRKADR
DAAA 21 52 FF       1613            LD HL,TRKMAP
DAAD 06 10          1614            LD B,TRK
DAAF C5             1615  PLALOOP   PUSH BC
DAB0 FD 36 01 40    1616            LD (IY+1),40H               ;NORMAL VELOCITY
DAB4 FD 36 02 58    1617            LD (IY+2),58H               ;ACCSENT 1 VELOCITY
DAB8 FD 36 03 70    1618            LD (IY+3),70H               ;ACCSENT 2 VELOCITY
DABC FD 36 08 20    1619            LD (IY+08H),20H             ;pp VOLUME
DAC0 FD 36 09 30    1620            LD (IY+09H),30H             ;p  VOLUME
DAC4 FD 36 0A 40    1621            LD (IY+0AH),40H             ;mp VOLUME
DAC8 FD 36 0B 50    1622            LD (IY+0BH),50H             ;mf VOLUME
DACC FD 36 0C 60    1623            LD (IY+0CH),60H             ;f  VOLUME
DAD0 FD 36 0D 70    1624            LD (IY+0DH),70H             ;ff VOLUME
DAD4 FD 36 1F 01    1625            LD (IY+1FH),1               ;TIMER RESET
DAD8 FD 36 20 07    1626            LD (IY+20H),7               ;GATE
DADC 3E 01          1627            LD A,1
DADE FD 77 1A       1628            LD (IY+1AH),A
DAE1 FD 77 1B       1629            LD (IY+1BH),A
DAE4 FD 77 1C       1630            LD (IY+1CH),A
DAE7 FD 77 1D       1631            LD (IY+1DH),A
DAEA 1A             1632            LD A,(DE)                   ;START ADR. (LOW)
DAEB FD 77 38       1633            LD (IY+38H),A
DAEE FD 77 36       1634            LD (IY+36H),A
DAF1 13             1635            INC DE
DAF2 1A             1636            LD A,(DE)                   ;START ADR. (HIGH)
DAF3 FD 77 39       1637            LD (IY+39H),A
DAF6 FD 77 37       1638            LD (IY+37H),A
DAF9 13             1639            INC DE
DAFA FD 36 3C 01    1640            LD (IY+3CH),01H             ;GATE FOR STACC
DAFE D5             1641            PUSH DE
DAFF E5             1642            PUSH HL
DB00 5E             1643            LD E,(HL)
DB01 16 00          1644            LD D,0
DB03 21 32 FF       1645            LD HL,USEMAP
DB06 19             1646            ADD HL,DE
DB07 FD E5          1647            PUSH IY
DB09 D1             1648            POP DE
DB0A E5             1649            PUSH HL
DB0B 21 78 00       1650            LD HL,0078H
DB0E 19             1651            ADD HL,DE
DB0F EB             1652            EX DE,HL
DB10 E1             1653            POP HL
DB11 01 05 00       1654            LD BC,05H
DB14 ED B0          1655            LDIR
DB16 11 90 00       1656            LD DE,TLEN
DB19 FD 19          1657            ADD IY,DE
DB1B E1             1658            POP HL
DB1C D1             1659            POP DE
DB1D 23             1660            INC HL
DB1E C1             1661            POP BC
DB1F 10 8E          1662            DJNZ PLALOOP
DB21 21 00 EE       1663            LD HL,TRKWORK
DB24 11 82 FF       1664            LD DE,MIDICH
DB27 06 10          1665            LD B,TRK
DB29 C5             1666  PLALP2    PUSH BC
DB2A 1A             1667            LD A,(DE)
DB2B E6 0F          1668            AND 0FH
DB2D 77             1669            LD (HL),A
DB2E 01 90 00       1670            LD BC,TLEN
DB31 09             1671            ADD HL,BC
DB32 13             1672            INC DE
DB33 C1             1673            POP BC
DB34 10 F3          1674            DJNZ PLALP2
DB36 3A 93 FF       1675            LD A,(SHIFT1)               ;TOTAL KEY SHIFT INIT
DB39 32 01 FF       1676            LD (0FF01H),A
DB3C 3E 78          1677            LD A,120
DB3E CD 96 E2       1678            CALL #STEMPO                ;SET TEMPO (START INT)
DB41 21 FF FF       1679            LD HL,0FFFFH
DB44 22 1E FF       1680            LD (0FF1EH),HL
DB47 E1             1681            POP HL
DB48 E5             1682            PUSH HL
DB49 5E             1683            LD E,(HL)
DB4A 23             1684            INC HL
DB4B 56             1685            LD D,(HL)
DB4C CD 7A DB       1686            CALL TRCHECK
DB4F D5             1687            PUSH DE
DB50 FD 21 00 EE    1688            LD IY,TRKWORK
DB54 06 10          1689            LD B,16
DB56 C5             1690  MILOOP    PUSH BC
DB57 CB 3A          1691            SRL D
DB59 CB 1B          1692            RR E
DB5B DC 9E DB       1693            CALL C,MDINIT
DB5E 01 90 00       1694            LD BC,TLEN
DB61 FD 09          1695            ADD IY,BC
DB63 C1             1696            POP BC
DB64 10 F0          1697            DJNZ MILOOP
DB66 D1             1698            POP DE
DB67 ED 53 0A FF    1699            LD (0FF0AH),DE              ;START
DB6B 7A             1700            LD A,D
DB6C B3             1701            OR E
DB6D 20 04          1702            JR NZ,PLSKIPX
DB6F 3E 01          1703            LD A,1                      ;STOP INT
DB71 D3 47          1704            OUT (CTC3),A
DB73 E1             1705  PLSKIPX   POP HL
DB74 FD E1          1706            POP IY
DB76 D1             1707            POP DE
DB77 C1             1708            POP BC
DB78 F1             1709            POP AF
DB79 C9             1710            RET
DB7A                1711
DB7A E5             1712  TRCHECK   PUSH HL
DB7B D5             1713            PUSH DE
DB7C C5             1714            PUSH BC
DB7D 11 00 00       1715            LD DE,0
DB80 21 61 FF       1716            LD HL,TRKMAP+TRK-1
DB83 06 10          1717            LD B,TRK
DB85 3E FE          1718            LD A,0FEH
DB87 BE             1719  TRCLOOP   CP (HL)
DB88 3F             1720            CCF
DB89 CB 13          1721            RL E
DB8B CB 12          1722            RL D
DB8D 2B             1723            DEC HL
DB8E 10 F7          1724            DJNZ TRCLOOP
DB90 EB             1725            EX DE,HL
DB91 C1             1726            POP BC
DB92 D1             1727            POP DE
DB93 7C             1728            LD A,H
DB94 A2             1729            AND D
DB95 57             1730            LD D,A
DB96 7D             1731            LD A,L
DB97 A3             1732            AND E
DB98 5F             1733            LD E,A
DB99 E1             1734            POP HL
DB9A 72             1735            LD (HL),D
DB9B 2B             1736            DEC HL
DB9C 73             1737            LD (HL),E
DB9D C9             1738            RET
DB9E                1739
DB9E D5             1740  MDINIT    PUSH DE
DB9F FD 7E 00       1741            LD A,(IY)
DBA2 4F             1742            LD C,A
DBA3 F6 B0          1743            OR 0B0H             ;COMTROL CANGE
DBA5 CD D6 E0       1744            CALL #OUT
DBA8 3E 01          1745            LD A,01H            ;MODULATION
DBAA CD D6 E0       1746            CALL #OUT
DBAD 3E 00          1747            LD A,0
DBAF FD 77 07       1748            LD (IY+07H),A
DBB2 CD D6 E0       1749            CALL #OUT
DBB5 3E 07          1750            LD A,07H            ;VOLUME
DBB7 CD D6 E0       1751            CALL #OUT
DBBA 3E 50          1752            LD A,50H
DBBC FD 77 04       1753            LD (IY+04H),A
DBBF CD D6 E0       1754            CALL #OUT
DBC2 3E 0A          1755            LD A,0AH            ;PANPOT
DBC4 CD D6 E0       1756            CALL #OUT
DBC7 3E 40          1757            LD A,40H
DBC9 FD 77 05       1758            LD (IY+05H),A
DBCC CD D6 E0       1759            CALL #OUT
DBCF 3E 0B          1760            LD A,0BH            ;EXPRESSION
DBD1 CD D6 E0       1761            CALL #OUT
DBD4 3E 7F          1762            LD A,7FH
DBD6 FD 77 06       1763            LD (IY+06H),A
DBD9 CD D6 E0       1764            CALL #OUT
DBDC 3E E0          1765            LD A,0E0H           ;BENDER
DBDE B1             1766            OR C
DBDF CD D6 E0       1767            CALL #OUT
DBE2 3E 00          1768            LD A,00H
DBE4 FD 77 0E       1769            LD (IY+0EH),A
DBE7 CD D6 E0       1770            CALL #OUT
DBEA 3E 40          1771            LD A,40H
DBEC FD 77 0F       1772            LD (IY+0FH),A
DBEF CD D6 E0       1773            CALL #OUT
DBF2 CD 54 E1       1774            CALL #ALLOUT
DBF5 D1             1775            POP DE
DBF6 C9             1776            RET
DBF7                1777
DBF7 18 18 02 E0    1778  SIOCD     DB 18H,18H,02H,0E0H,04H,44H,03H,0C1H,05H,68H
DBFB 04 44 03 C1
DBFF 05 68
DC01                1779
DC01 F5             1780  CONT      PUSH AF
DC02 E5             1781            PUSH HL
DC03 D5             1782            PUSH DE
DC04 C5             1783            PUSH BC
DC05 3A 30 FF       1784            LD A,(OPENF)
DC08 FE 55          1785            CP 55H
DC0A 20 06          1786            JR NZ,CNSKIP                ;Not open
DC0C 3A 00 FF       1787            LD A,(0FF00H)
DC0F CD 96 E2       1788            CALL #STEMPO
DC12 C1             1789  CNSKIP    POP BC
DC13 D1             1790            POP DE
```

```
DC14 E1           1791          POP HL
DC15 F1           1792          POP AF
DC16 C9           1793          RET
DC17              1794
DC17 F5           1795  STOP    PUSH AF
DC18 E5           1796          PUSH HL
DC19 D5           1797          PUSH DE
DC1A C5           1798          PUSH BC
DC1B 3A 30 FF     1799          LD A,(OPENF)
DC1E FE 55        1800          CP 55H
DC20 CC 28 DC     1801          CALL Z,STOPX
DC23 C1           1802          POP BC
DC24 D1           1803          POP DE
DC25 E1           1804          POP HL
DC26 F1           1805          POP AF
DC27 C9           1806          RET
DC28              1807
DC28 3E 01        1808  STOPX   LD A,01H
DC2A D3 47        1809          OUT (CTC3),A
DC2C AF           1810          XOR A
DC2D 32 6E E1     1811          LD (#EXF),A
DC30 21 00 F7     1812          LD HL,MIDIKEY
DC33 11 01 F7     1813          LD DE,MIDIKEY+1
DC36 01 FF 07     1814          LD BC,MIDIX-1
DC39 77           1815          LD (HL),A
```

```
DC3A ED B0        1816          LDIR
DC3C 06 10        1817          LD B,TRK
DC3E 3E B0        1818  STPLOOP LD A,0B0H
DC40 48           1819          LD C,B
DC41 0D           1820          DEC C
DC42 B1           1821          OR C
DC43 CD D6 E0     1822          CALL #OUT
DC46 3E 7B        1823          LD A,7BH
DC48 CD D6 E0     1824          CALL #OUT
DC4B AF           1825          XOR A
DC4C CD D6 E0     1826          CALL #OUT
DC4F 10 ED        1827          DJNZ STPLOOP
DC51 CD 54 E1     1828          CALL #ALLOUT
DC54 C9           1829          RET
DC55              1830
DC55 F5           1831  SPSKIP  PUSH AF
DC56 1A           1832  SKPLOOP LD A,(DE)
DC57 FE 20        1833          CP 20H
DC59 20 03        1834          JR NZ,SKIPEND
DC5B 13           1835          INC DE
DC5C 18 F8        1836          JR SKPLOOP
DC5E F1           1837  SKIPEND POP AF
DC5F C9           1838          RET
DC60              1839
```

リスト7　KENBANソースリスト

```
0000                 1          ;***** KENBAN INDICATOR FOR PLAYER.X *****
C000                 2          ORG 0C000H
F700 P               3  MIDI    EQU 0F700H
C800 P               4  KEN     EQU 0C800H
FF00 P               5  #SYSTEM EQU 0FF00H
C000                 6
C000 C3 41 C0        7          JP INIT
C003 C3 6A C0        8          JP MAIN
C006 C3 22 C1        9          JP AUTO
C009                10
C009                11
C009 01             12  TIMER   DB 1
C00A 01             13  CONST   DB 1
C00B 07 06 05 04    14  COLOR   DB 7,6,5,4,3,2,1,7,6,5,4,3,2,1,7,6
C00F 03 02 01 07
C013 06 05 04 03
C017 02 01 07 06
C01B                15
C01B                16          ;*** SET MAPPING ***
C01B F5             17  MAPSET  PUSH AF
C01C 3E 02          18          LD A,2
C01E F3             19          DI
C01F D3 B4          20          OUT (0B4H),A
C021 3A B5 00       21          LD A,(0B5H)
C024 32 40 C0       22          LD (MAPBUF),A
C027 3E 02          23          LD A,2
C029 D3 B4          24          OUT (0B4H),A
C02B 3E 38          25  MAPNO   LD A,38H
C02D D3 B5          26          OUT (0B5H),A
C02F FB             27          EI
C030 F1             28          POP AF
C031 C9             29          RET
C032                30
C032                31          ;*** RESET MAPPING ***
C032 F5             32  MAPRES  PUSH AF
C033 3E 02          33          LD A,2
C035 F3             34          DI
C036 D3 B4          35          OUT (0B4H),A
C038 3A 40 C0       36          LD A,(MAPBUF)
C03B D3 B5          37          OUT (0B5H),A
C03D FB             38          EI
C03E F1             39          POP AF
C03F C9             40          RET
C040                41
C040                42  MAPBUF DS 1
C041                43
C041                44          ;*** INIT BUFF ***
C041 F5             45  INIT    PUSH AF
C042 E5             46          PUSH HL
C043 D5             47          PUSH DE
C044 C5             48          PUSH BC
C045 21 00 C8       49          LD HL,KEN
C048 11 01 C8       50          LD DE,KEN+1
C04B 01 FF 07       51          LD BC,07FFH
C04E 36 00          52          LD (HL),0
C050 ED B0          53          LDIR
C052 3E 01          54          LD A,1
C054 32 09 C0       55          LD (TIMER),A
C057 3E 39          56          LD A,39H
C059 32 2C C0       57          LD (MAPNO+1),A
C05C CD 1B C0       58          CALL MAPSET
C05F CD 00 C1       59          CALL PCGSET
C062 CD 32 C0       60          CALL MAPRES
C065 C1             61          POP BC
C066 D1             62          POP DE
C067 E1             63          POP HL
C068 F1             64          POP AF
C069 C9             65          RET
C06A                66
C06A                67          ;*** MAIN ***
C06A F5             68  MAIN    PUSH AF
C06B E5             69          PUSH HL
C06C D5             70          PUSH DE
C06D C5             71          PUSH BC
C06E D9             72          EXX
C06F E5             73          PUSH HL
C070 D5             74          PUSH DE
C071 C5             75          PUSH BC
C072 D9             76          EXX
C073 3A 09 C0       77          LD A,(TIMER)
C076 3D             78          DEC A
C077 32 09 C0       79          LD (TIMER),A
C07A 20 42          80          JR NZ,END
C07C 3A 0A C0       81          LD A,(CONST)
C07F 32 09 C0       82          LD (TIMER),A
C082 3E 38          83          LD A,38H
C084 32 2C C0       84          LD (MAPNO+1),A
C087 CD 1B C0       85          CALL MAPSET
C08A D9             86          EXX
C08B 21 18 F7       87          LD HL,MIDI+24
C08E 11 18 C8       88          LD DE,KEN+24
C091 D9             89          EXX
C092 21 90 41       90          LD HL,80*5+4000H        ;START LINE=5
C095 11 0B C0       91          LD DE,COLOR
C098 06 10          92          LD B,16
C09A C5             93  LOOP    PUSH BC
C09B D5             94          PUSH DE
C09C E5             95          PUSH HL
C09D 1A             96          LD A,(DE)
C09E E6 07          97          AND 07H
C0A0 F6 10          98          OR 10H
C0A2 32 F5 C0       99          LD (COLX+1),A
C0A5 CD C8 C0      100          CALL PRT1CH
C0A8 D9            101          EXX
C0A9 01 80 00      102          LD BC,80H
C0AC 09            103          ADD HL,BC
C0AD EB            104          EX DE,HL
C0AE 09            105          ADD HL,BC
C0AF EB            106          EX DE,HL
C0B0 D9            107          EXX
C0B1 E1            108          POP HL
C0B2 01 50 00      109          LD BC,80
C0B5 09            110          ADD HL,BC
C0B6 D1            111          POP DE
C0B7 13            112          INC DE
C0B8 C1            113          POP BC
C0B9 10 DF         114          DJNZ LOOP
C0BB               115
C0BB CD 32 C0      116          CALL MAPRES
C0BE               117
C0BE D9            118  END     EXX
C0BF C1            119          POP BC
C0C0 D1            120          POP DE
C0C1 E1            121          POP HL
C0C2 D9            122          EXX
C0C3 C1            123          POP BC
C0C4 D1            124          POP DE
C0C5 E1            125          POP HL
C0C6 F1            126          POP AF
C0C7 C9            127          RET
C0C8               128
C0C8 06 50         129  PRT1CH  LD B,80
C0CA D9            130          EXX
C0CB E5            131          PUSH HL
C0CC D5            132          PUSH DE
C0CD D9            133          EXX
C0CE D9            134  PLOOP   EXX
C0CF 1A            135          LD A,(DE)
C0D0 FE 05         136          CP 5
C0D2 38 02         137          JR C,PSKIP1
C0D4 3D            138          DEC A
C0D5 12            139          LD (DE),A
C0D6 F3            140  PSKIP1  DI
C0D7 4E            141          LD C,(HL)
C0D8 79            142          LD A,C
C0D9 E6 7F         143          AND 7FH
C0DB 77            144          LD (HL),A
C0DC FB            145          EI
C0DD EB            146          EX DE,HL
C0DE 28 09         147          JR Z,PSKIP2
C0E0 79            148          LD A,C
C0E1 E6 80         149          AND 80H
C0E3 28 06         150          JR Z,PNEXT1
C0E5 36 0F         151          LD (HL),0FH
C0E7 18 02         152          JR PNEXT1
C0E9 36 00         153  PSKIP2  LD (HL),0
C0EB 7E            154  PNEXT1  LD A,(HL)
C0EC EB            155          EX DE,HL
C0ED 23            156          INC HL
C0EE 13            157          INC DE
C0EF D9            158          EXX
C0F0 87            159          ADD A,A
C0F1 77            160          LD (HL),A
C0F2 CB DC         161          SET 3,H
C0F4 36 16         162  COLX    LD (HL),16H               ;PCG=1 COLOR=6
C0F6 CB 9C         163          RES 3,H
C0F8 23            164          INC HL
C0F9 10 D3         165          DJNZ PLOOP
C0FB D9            166          EXX
C0FC D1            167          POP DE
C0FD E1            168          POP HL
C0FE D9            169          EXX
C0FF C9            170          RET
C100               171
C100               172
C100 21 00 48      173  PCGSET  LD HL,4800H               ;PCG1 CODE:00H-1FH
C103 06 10         174          LD B,16
C105 C5            175  PSLOOP  PUSH BC
C106 05            176          DEC B
C107 28 05         177          JR Z,PSSKIP1
C109 36 00         178  PSLOOP2 LD (HL),0
C10B 23            179          INC HL
C10C 10 FB         180          DJNZ PSLOOP2
C10E C1            181  PSSKIP1 POP BC
C10F C5            182          PUSH BC
C110 3E 10         183          LD A,16
C112 90            184          SUB B
C113 28 06         185          JR Z,PSSKIP2
C115 47            186          LD B,A
C116 36 7E         187  PSLOOP3 LD (HL),7EH
C118 23            188          INC HL
C119 10 FB         189          DJNZ PSLOOP3
C11B 36 00         190  PSSKIP2 LD (HL),0
C11D 23            191          INC HL
C11E C1            192          POP BC
C11F 10 E4         193          DJNZ PSLOOP
C121 C9            194          RET
C122               195
C122 F5            196  AUTO    PUSH AF
C123 E5            197          PUSH HL
C124 DF            198  AUTOLP  RST 18H          ;CHECK SHIFT+BREAK
C125 0E            199          DB 0EH
C126 28 0C         200          JR Z,AUTOEND
C128 2A 0A FF      201          LD HL,(#SYSTEM+0AH)
C12B 7C            202          LD A,H
C12C B5            203          OR L
C12D 28 05         204          JR Z,AUTOEND
C12F CD 6A C0      205          CALL MAIN
C132 18 F0         206          JR AUTOLP
C134 CD 6A C0      207  AUTOEND CALL MAIN
C137 E1            208          POP HL
C138 F1            209          POP AF
C139 C9            210          RET
```

THE SENTINEL

●8ビットの世界を席巻したOS

パソコンがまだマイコンと呼ばれていた8ビットマシンの全盛期に，i8080というCPU（Z80の前身）を対象としたOSが大流行した時期がありました。OSというよりDOSといったほうが正しいこのシステムこそ，CP/Mです。世界中のi8080マシン，Z80マシンに載り，多彩なアプリケーションを生み出しました。言語，表集計ソフト，ワープロ，そしてゲームと，他の追随を許さないソフトウェア資産がCP/M上で蓄積されたのです。現在パソコン用OSでもっとも普及しているMS-DOSの最初のバージョンが，8086版のCP/MであるCP/M-86に毛の生えた程度のものでしかなかったということからも，当時のCP/Mの勢いをうかがい知ることができます。

S-OS用のアプリケーションも，多かれ少なかれCP/Mを意識して開発されているといえるでしょう。初期のうちにあれほどの言語処理系がサポートされたのも，CP/Mへの対抗意識のあらわれといえるのではないでしょうか。

パソコンが16ビットのビジネスマシン主導になってしまい，MS-DOSがここまで普及してしまった現在，膨大な遺産を抱えたままCP/Mは眠りに就こうとしています。

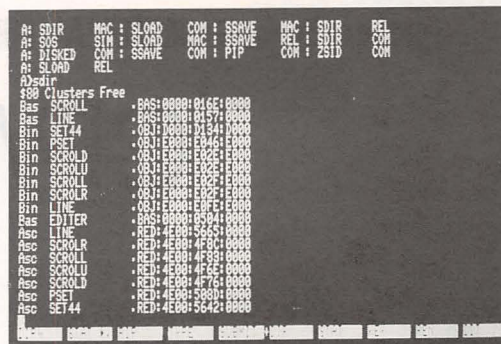
第83部
CP/M用ファイルコンバータ

●CP/MとS-OSのファイルコンバータ

CP/Mとともに，数多くのよくできたPDS(Public Domain Software)を眠りに就かせてしまっていいものでしょうか。「いまや8ビットパソコン用のOSとしてNo.1の実働率を誇る(と自負している)S-OSで，これらのプログラムに再び命を与えよう」と，今回お届けするのはCP/MとS-OSのファイルを相互に変換するファイルコンバータです。

CP/Mのディスク管理システムの性格上，S-OS用のひとつのプログラムですべてのCP/Mのディスクを扱うのは難しいため，このコーナーとしては異色ではありますが，このコンバータはCP/M上で動作するプログラムとなっています。

もともと，X1用のCP/Mではテープを介してCP/MとHuBASICでデータをやりとりできましたが，このコンバータによってディスクからディスクへ直接データを転送できるようになりました。テープなどがサポートされていないX1以外の機種では，ようやくCP/MとS-OSのデータの互換が取れるようになったわけです。

この機会にもう一度CP/Mを復活させ，手持ちのプログラムに新しい命を与えてみてはいかがでしょうか。

●S-OSの系譜(2)

MZ-2200で動いているマシン語プログラムが，テープ交換だけで自分のX1でも実行できる。このような環境を提供することを目的として1985年6月号からS-OSはスタートしました。この最初のバージョンはS-OS“MACE”(棍棒)と命名されました。MACEは多くのロールプレイングゲームで最初に与えられる武器です。“AXE”，“SWORD”，と成長させようという願いを込めての命名でした。

同じZ80を使っていながら機種によってプログラムが異なるのは，画面に文字を出力したり，キーボードから文字を入力する方法が機種によって異なっているからです。S-OSはこれらの入出力の方法を統一し，どの機種でも同じ方法で画面に文字を出せるように，同じ方法でキー入力できるようにしたのです。S-OSが提供する入出力だけを使えば，S-OS搭載のすべてのマシンでまったく同じプログラムが走るようになるわけです。

皆さんに入力してもらわなければならないダンプリストの量や，開発にかかる時間を考えて，このS-OS“MACE”は各機種に用意されているモニタやIOCS(Input Output Control System)がすでに持っている機能を利用して作られています。

プログラムがS-OSの提供する入出力を利用すると，S-OSはモニタやIOCSが持っているルーチンを使って処理を行います。ただしモニタやIOCSのルーチンは微妙に処理内容が異なったり，S-OSでは保存すべきレジスタ内容を破壊するものがあります。S-OSはそのあたりの調整を行って，同じ条件で入出力を利用できるようにしています。

またS-OSは各機種で異なる文字コードを統一し，同じコードで同じ文字が表示されるようにしています。“MACE”で使用できるのは英大文字/小文字，数字，記号，そして画面クリアと改行の2つのコントロールコードだけでした。

いろいろ制限はあったものの，他機種と同じプログラムを利用できる環境は整いました。“MACE”と同時に発表されたLisp85はたった1本のダンプリストで全機種に対応したのです。

全機種共通CP/M-80要

CP/M用ファイルコンバータ

CP/M用のPDSなどをS-OSに持ってくる，S-OSのアプリケーションをCP/Mで使う。こんなことができれば，使えるソフトウェアの種類は大きく広がります。そこでファイルコンバータ。FuzzyBASICコンパイラの石上君の作品です。

Ishigami Tatuya
石上 達也

ファイルコンバータとはなにか

世の中には星の数ほどパソコンやそのシステムがあり，それぞれが独自にディスクへの書き込み方法を持っています。少しでも速く，無駄がないようにということなのですが，これはそれぞれのシステム間でデータのやりとりができなくなってしまうというデメリットがあります。

われらがS-OSのシステムを見てみると，X1のファイルフォーマットに似た方法で管理されています。S-OS上のみでソフト開発をしている分には問題はないのですが，私はCP/M上の「Small-C」というプログラムをぜひS-OS“SWORD”に移植したいと思い立ったのです。

CP/Mはいまはもう昔のシステムで，ファイル名に漢字はおろか，英小文字すら使えないというしろものです。ファイルもS-OSとは似ても似つかない方法で管理されています。しかし，歴史があるということは，使っている人も多く(?)，素晴らしいプログラムも多いということです。よくできたPDS(Public Domain Software)など，CP/MからS-OS“SWORD”へ持ってきたいものはいっぱいあるだろうと考えて，CP/Mのディスクから S-OSのディスクヘデータやプログラムを転送するプログラム「ファイルコンバータ」を作りました。

なお，このプログラムを使用するにはCP/Mが必要です。

プログラムの使い方

まず，各自のCP/Mを起動します。そして，CP/Mのディスクを Aドライブ，S-OSのディスクをBドライブにセットしてください。以下に説明する操作はすべて，ディスクがこのようにセットされている状態で行います。

AドライブにセットするCP/Mのディスクには，プログラム1，2，3のうち必要なものをあらかじめPIPコマンドで転送しておいてください。また，ログインドライブはAドライブにしてください（要は，リブートしたときにS-OSのディスクへCCPを読みに行かなければいい）。

ディレクトリを見る

SDIR B

ドライブにセットされているS-OS用のディスクのディレクトリを見ることができます。

S-OS→CP/M

SLOAD〔ファイル名〕

S-OSのファイルを読み込み，〔ファイル名〕という名のCP/Mファイルにします。

SLOADコマンドを実行すると，

File mode 0……Bin 1……Asc

と画面に表示されます。転送したいS-OSのファイルがバイナリ形式なら0，ASCII形式なら1を入力してください。次に

File name：

と画面に表示されるので，転送したいS-OS上のファイル名を入力してください。しばらくAドライブとBドライブが動いて，ファイルの転送が行われます。

S-OSにはFuzzyBASICのテキストファイルであるBas形式がありますが，これはCP/Mに転送しても無意味なのでサポートしていません。必要なら，Bas形式のファイルを一度Asc形式のファイルに変換してから転送してください（変換方法はOh!MZ1986年1月号，「FuzzyBASIC料理法<2>」を参照)。

CP/M→S-OS

SSAVE〔ファイル名〕

CP/Mの〔ファイル名〕という名のファイルをS-OSのディスクへ転送します。

SLOADコマンドと同じようにBin形式かAsc形式かを選択し，次の，

File neme：

では，ファイルをS-OSのディスクに転送したときに与えるファイル名を入力します。

転送するファイルがBin形式のときはさらに，

Start Address：

Exec Address：

と表示されますので，それぞれを4桁の16進数で入力してください。ここで入力するStart/Exec Addressは，S-OSのディスクに転送したときに意味をもつアドレスです。CP/MでS-OSのプログラムを開発する場合などに使います。転送が終わると，

File Size：XXXX

と表示されて完了です。

プログラムの入力方法

まず，リスト4をDDT，ZSID（あればS-BUGでもかまわない）を使って入力してください。これはチェックサム出力用プログラムですので，まず自分自身のチェックサムを確認して，そこで間違いがないようならば，同様にリスト1，2，3をそれぞれ入力してください。

プログラムについて

このプログラムはS-OS上ではなく，CP/M上で動きます。SENTINELのコーナーになぜCP/Mでなければ動かないプログラムかというと，

1）SWORDはすべてソースレベルで公開されているので，同じ動作をするサブルーチンを簡単にCP/M上に用意してアプリケーションを実行することができる。これに対してCP/Mは内部が公開されていないので，逆のことをやろうとすると面倒である。

2）CP/Mは内部のワークエリアを変更することにより，1トラックのセクタ数やトラックの最初のセクタ番号を勝手に変えることができる。このため，X1のCP/MとS-OSのファイルコンバータを作ったからといって，それがすべてのCP/Mディスクに対して使用できるとは限らない。

という2つの理由があります。

このプログラムはまず，起動されたCP/

Mシステムがどのようにディスクを扱っているのかを調べ，それに合わせて動作する，いわば「郷に入りては郷に従う」プログラムにしてあります。

S-OSのディスク周りのサブルーチンは，SWORDのDOSモジュールから借りてきたものばかりです。あとはCP/MのBDOSコールの定石しか使っていないので，プログラムを読むのは簡単でしょう。この方面に興味のある方には，『応用CP/M』（村田康治著，アスキー出版）を紹介しておきます。この本は，このプログラムを作るのに私も大いに参考にした書籍です。

今後の予定

長い受験生活を終え，ようやく私も都内の私立大学へ通うことができるようになりました。この約10カ月，パソコン関係の雑誌にほとんど触れていなかったので，ちょっとした浦島太郎気分です。

ひと昔前までは，1年もたてばアッというようなものが出ていたように思うのですが，この1年間の空白を埋めてみると，意外と進歩していないのにガッカリしました。それとともに，8ビットの世界から抜け出せない私としては少しばかり安心してもいます。

そんなわけで，当分の間はこの「THE SENTINEL」のコーナーにプログラムを発表していこうと思っています。

移植の完了した“Small-C”を使えるように，現在リロケータブルアセンブラを作成中です。これが完成すればS-OSでCコンパイラを使えるようになるわけです。ご期待ください。

リスト1 SDIR

```
ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum
0100 C3 02 07 0E 1F CD 05 00:CB
0108 7E 23 66 6F 22 F1 02 ED:78
0110 5B 01 00 1B 1B 1B 21 1B:E9
0118 00 19 22 4D 01 21 1E 00:C8
0120 19 22 57 01 21 21 00 19:EE
0128 22 5A 01 21 24 00 19 22:FD
0130 5D 01 21 27 00 19 22 60:41
0138 01 21 2A 00 19 22 69 01:F1
0140 0E 01 CD 4C 01 D8 36 00:37
0148 23 36 00 C9 CD 00 00 7C:6B
0150 B5 C0 3E 02 37 C9 C3 00:78
0158 00 C3 00 00 C3 00 00 CD:53
0160 00 00 A7 C8 3E 01 37 C9:AE
0168 CD 00 00 A7 C8 3E 01 37:B2
0170 C9 E5 D5 C5 0E 06 1E FF:79
0178 CD 05 00 FE 03 CA 00 00:9D
-----------------------------------
SUM: 7E 81 B9 77 9A 06 39 EC 149C

ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum
0180 C1 D1 E1 C9 E5 D5 C5 F5:B0
0188 0E 02 5F CD 05 00 F1 C1:F3
0190 D1 E1 C9 F5 0F 0F 0F 0F:AC
0198 CD 9C 01 F1 E6 0F C6 30:46
01A0 FE 3A 38 02 C6 07 CD 84:90
01A8 01 C9 7C CD 93 01 7D CD:F1
01B0 93 01 C9 CD BC 01 67 D4:22
01B8 BC 01 6F C9 C5 1A 13 CD:B4
01C0 D3 01 38 0D 0F 0F 0F 0F:55
01C8 4F 1A 13 CD D3 01 38 01:56
01D0 B1 C1 C9 D6 30 38 0F FE:86
01D8 0A 38 08 FE 11 38 07 D6:6E
01E0 07 FE 10 37 3F C9 3E 0E:A0
01E8 37 C9 F5 3E 0A CD 84 01:8F
01F0 3E 0D CD 84 01 F1 C9 F5:4C
01F8 3E 20 CD 84 01 F1 C9 D5:3F
-----------------------------------
SUM: 52 5D B1 0C 27 0E 00 A4 F29C

ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum
0200 F5 1A A7 28 06 CD 84 01:36
0208 13 18 F6 F1 D1 C9 E5 D5:66
0210 C5 0E 01 CD 05 00 C1 D1:38
0218 E1 C9 E5 D5 C5 0E 0A 11:52
0220 DB 02 CD 05 00 3A DC 02:C7
0228 21 DD 02 16 00 5F 19 36:C4
0230 00 C1 D1 E1 C9 CD 4A 02:55
0238 D5 21 B6 02 11 96 02 01:58
0240 12 00 ED B0 D1 CD C9 02:18
0248 B7 C9 21 B6 02 77 23 06:F9
0250 0D CD 7F 02 1A 20 03 3E:D6
0258 20 1B FE 2E 20 03 3E 20:E8
0260 1B 77 13 23 10 EB 1A FE:DB
0268 2E 20 01 13 06 03 CD 7F:B7
0270 02 1A 20 03 3E 20 1B 77:2F
0278 13 23 10 F2 36 20 C9 D5:2C
-----------------------------------
SUM: D3 4F A8 7A 12 35 6D 22 6BD0

ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum
0280 CD C9 02 1A D1 FE 3A C8:83
0288 FE 20 D0 BF C9 FE 61 D8:AD
0290 FE 7B D0 D6 20 C9 00 00:08
0298 00 00 00 00 00 00 00 00:00
02A0 00 00 00 00 00 00 00 00:00
02A8 00 00 00 00 00 00 00 00:00
02B0 00 00 00 00 00 00 00 00:00
02B8 00 00 00 00 00 00 00 00:00
02C0 00 00 00 00 00 00 00 00:00
02C8 13 1A FE 20 28 FA C9 96:CC
02D0 02 02 05 02 06 10 00 0E:2F
02D8 00 DD 02 14 00 00 00 00:F3
02E0 00 00 00 00 00 00 00 00:00
02E8 00 00 00 00 00 00 00 00:00
02F0 00 00 00 E5 D5 C5 F5 0E:82
02F8 01 CD 4C 01 38 21 F1 C1:26
-----------------------------------
SUM: DF 2A F3 CB F5 B5 4A 13 1956

ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum
0300 D1 E1 E5 D5 C5 F5 D5 E5:E0
0308 CD 27 03 38 10 E1 11 00:31
0310 01 19 D1 13 F1 3D 20 ED:39
0318 A7 C1 D1 E1 C9 E1 D1 F1:86
0320 C1 D1 E1 3E 02 37 C9 D5:88
0328 22 FD 03 44 4D CD 5C 01:DD
0330 D1 EB 29 ED 5B F1 02 CD:ED
0338 DB 03 D5 44 4D CD 56 01:68
0340 D1 42 4B ED 43 FF 03 CD:5D
0348 59 01 CD 5F 01 D8 2A FD:86
0350 03 11 80 00 19 44 4D CD:0B
0358 5C 01 ED 4B FF 03 03 CD:67
0360 59 01 CD 5F 01 C9 E5 D5:0A
0368 C5 F5 0E 01 CD 4C 01 7C:5F
0370 B5 28 21 F1 C1 D1 E1 E5:47
0378 D5 C5 F5 D5 E5 CD 9C 03:B5
-----------------------------------
SUM: 06 D6 E2 71 56 87 34 04 0442

ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum
0380 38 10 E1 11 00 01 19 D1:25
0388 13 F1 3D 20 ED A7 C1 D1:87
0390 E1 C9 E1 D1 F1 C1 D1 E1:C0
0398 3E 02 37 C9 D5 22 FD 03:37
03A0 44 4D CD 5C 01 D1 EB 29:A0
03A8 ED 5B F1 02 CD DB 03 D5:BB
03B0 44 4D CD 56 01 D1 42 4B:13
03B8 ED 43 FF 03 CD 59 01 CD:26
03C0 68 01 D8 2A FD 03 11 80:FC
03C8 00 19 44 4D CD 5C 01 ED:C1
03D0 4B FF 03 03 CD 59 01 CD:44
03D8 68 01 C9 F5 C5 42 4B 54:CD
03E0 5D 3E 10 21 00 00 CB 23:BA
03E8 CB 12 ED 6A E5 B7 ED 42:FF
03F0 E1 38 03 ED 42 13 3D 20:BB
03F8 ED EB C1 F1 C9 00 00 00:53
-----------------------------------
SUM: DD 91 69 5A 9B 25 2C AF 5D61

ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum
0400 00 21 16 04 87 16 00 5F:37
0408 19 5E 23 56 CD FF 01 3E:FB
0410 07 CD 84 01 CD EA 01 C9:DA
0418 34 04 45 04 54 04 68 04:45
0420 78 04 83 04 91 04 A6 04:42
0428 B5 04 C1 04 D5 04 E6 04:41
0430 F4 04 FB 04 44 65 76 69:7F
0438 63 65 20 49 2F 4F 20 45:14
0440 72 72 6F 72 00 44 65 76:E4
0448 69 63 65 20 6F 66 66 6C:F8
0450 69 6E 65 00 42 61 64 20:63
0458 66 69 6C 65 20 44 65 73:DC
0460 63 72 69 70 74 65 72 00:F9
0468 57 72 69 74 65 20 50 72:ED
0470 6F 74 65 63 74 65 64 00:E8
0478 42 61 64 20 52 65 63 6F:B0
-----------------------------------
SUM: ED 26 A1 12 BE 5D A9 76 8BC5

ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum
0480 72 64 00 42 61 64 20 66:63
0488 69 6C 65 20 6D 6F 64 65:FF
0490 00 42 61 64 20 41 6C 6C:40
0498 6F 63 61 74 69 6F 6E 20:0D
04A0 54 61 62 6C 65 00 46 69:97
04A8 6C 65 20 6E 6F 74 20 46:A8
04B0 6F 75 6E 64 00 44 65 76:D5
04B8 69 63 72 20 46 75 6C 6C:F1
04C0 00 46 69 6C 65 20 41 6C:4D
04C8 72 65 61 64 79 20 45 78:F2
04D0 69 73 74 73 00 52 65 73:ED
04D8 65 72 76 65 64 20 46 65:E1
04E0 61 74 75 72 65 00 46 69:D0
04E8 6C 65 20 6E 6F 74 20 4F:B1
04F0 70 65 6E 00 53 79 6E 74:F1
04F8 61 78 20 45 72 72 6F 72:03
-----------------------------------
SUM: C0 59 60 65 4C C1 09 42 77FD

ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum
0500 20 00 00 00 00 00 00 00:20
0508 00 00 00 00 00 00 00 00:00
0510 00 00 00 00 00 00 00 00:00
0518 00 00 00 00 00 00 00 00:00
0520 00 00 00 00 00 00 00 00:00
0528 00 00 00 00 00 00 00 00:00
0530 00 00 00 00 00 00 00 00:00
0538 00 00 00 00 00 00 00 00:00
0540 00 00 00 00 00 00 00 00:00
0548 00 00 00 00 00 00 00 00:00
0550 00 00 00 00 00 00 00 00:00
0558 00 00 00 00 00 00 00 00:00
0560 00 00 00 00 00 00 00 00:00
0568 00 00 00 00 00 00 00 00:00
0570 00 00 00 00 00 00 00 00:00
0578 00 00 00 00 00 00 00 00:00
-----------------------------------
SUM: 20 00 00 00 00 00 00 00 8EC8
```

（$06FF_H$まで00_Hで埋める）

```
ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum
0700 00 00 31 00 40 CD 03 01:42
0708 CD 11 07 DC 01 04 C3 00:89
0710 00 CD FF 07 D8 3E 24 CD:DA
0718 84 01 CD EA 07 CD 93 01:A4
0720 11 34 08 CD FF 01 06 10:30
0728 ED 5B D5 02 21 02 05 3E:85
0730 01 CD F3 02 D8 CD 3E 07:AD
0738 28 03 13 10 EF C9 C5 D5:A0
0740 06 08 7E A7 28 0D FE FF:65
0748 28 10 CD 5F 07 CD EA 01:23
0750 CD 71 01 11 20 00 19 10:99
0758 E9 3E AF D1 C1 B7 C9 C5:AD
0760 D5 E5 11 45 08 01 20 00:39
0768 ED B0 CD 92 07 CD BD 07:94
0770 2A 59 08 CD 89 07 ED 5B:30
0778 57 08 19 2B CD 89 07 2A:2A
-----------------------------------
SUM: 9F FB E1 65 7C 64 26 5A 16A7

ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum
0780 5B 08 CD 89 07 E1 D1 C1:33
0788 C9 3E 3A CD 84 01 CD AA:0A
0790 01 C9 3A 45 08 CB 7F 28:C3
0798 03 3E 08 11 E6 07 6F 26:DC
07A0 00 29 29 11 10 08 19 EB:7F
07A8 CD FF 01 3A 45 08 CB 77:96
07B0 3E 2A 20 02 3E 20 CD 84:39
07B8 01 CD F7 01 C9 11 46 08:EE
07C0 06 0D 1A FE 20 30 02 3E:BB
07C8 20 FE 2E 20 02 3E 20 CD:99
07D0 84 01 13 10 ED 3E 2E CD:CE
07D8 84 01 06 03 1A FE 20 30:F6
07E0 02 3E 20 CD 84 01 13 10:D5
07E8 F3 C9 C5 E5 06 80 0E 00:FA
07F0 2A D3 02 7E A7 20 01 0C:51
07F8 23 10 F8 79 E1 C1 C9 D5:E4
-----------------------------------
SUM: A4 63 CA D4 10 01 DE A0 3E65

ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum
0800 E5 ED 5B D7 02 2A D3 02:05
0808 3E 01 CD F3 02 E1 D1 C9:7C
0810 4E 75 6C 00 42 69 6E 00:48
0818 42 61 73 00 3F 3F 3F 00:D3
0820 41 73 63 00 3F 3F 3F 00:D4
0828 3F 3F 3F 00 3F 3F 3F 00:7A
0830 44 69 72 00 20 43 6C 75:63
0838 73 74 65 72 73 20 46 72:09
0840 65 65 0D 0A 00 00 00 00:E1
0848 00 00 00 00 00 00 00 00:00
0850 00 00 00 00 00 00 00 00:00
0858 00 00 00 00 00 00 00 00:00
0860 00 00 00 00 00 01 13 10:24
0868 F3 C9 C5 E5 06 80 0E 00:FA
0870 2A D3 02 7E A7 20 01 0C:51
0878 23 10 F8 79 E1 C1 C9 D5:E4
-----------------------------------
SUM: 8F 64 4C 22 24 F6 6C A3 E5EB
```

リスト2 SSSAVE

(06FF$_H$までリスト1と同じ)

```
ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum
0700 00 00 31 00 40 CD 03 01:42
0708 CD 11 07 DC 01 04 C3 00:89
0710 00 11 20 08 CD FF 01 CD:D3
0718 0E 02 06 01 FE 30 28 02:6F
0720 06 04 78 32 73 08 CD EA:E6
0728 01 11 3B 08 CD FF 01 ED:0F
0730 5B D9 02 CD 1A 02 CD EA:D6
0738 01 AF 32 FA 09 32 74 08:93
0740 3E 01 32 5C 00 0E 00 CD:A8
0748 4C 01 D8 0E 0F 11 5C 00:AF
0750 CD 05 00 FE FF 3E 08 37:4C
0758 C8 3A 73 08 ED 5B D9 02:A0
0760 CD 76 08 D8 AF 32 7C 00:80
0768 3A 73 08 FE 04 28 3D 11:2D
0770 46 08 CD FF 01 ED 5B D9:3C
0778 02 CD 1A 02 CD EA 01 CD:70
-------------------------------------
SUM: AC C0 B9 2D EB 24 50 56 E36A

ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum
0780 B3 01 22 09 0B D8 11 55:28
0788 08 CD FF 01 ED 5B D9 02:F8
0790 CD 1A 02 CD EA 01 CD B3:21
0798 01 22 0B 0B D8 CD D9 07:BE
07A0 CD D1 09 D8 3A 75 08 A7:DD
07A8 28 F3 18 18 CD D9 07 FE:F6
07B0 0A 28 F9 FE 1A 28 09 CD:41
07B8 D1 09 3A 75 08 A7 28 EC:4C
07C0 AF CD D1 09 11 64 08 CD:A0
07C8 FF 01 2A 07 0B CD AA 01:B4
07D0 2C 2D C4 FB 09 CD A3 08:99
07D8 C9 AF 32 75 08 3A 74 08:DD
07E0 E6 7F F5 CC 06 08 F1 C6:EB
07E8 80 26 00 6F 7E F5 3A 74:36
07F0 08 3C E6 7F 32 74 08 20:77
07F8 0B 21 75 08 7E 36 00 A7:04
-------------------------------------
SUM: 75 AB C3 87 44 FD CC 4E 7E1F

ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum
0800 28 02 36 01 F1 C9 0E 00:29
0808 CD 4C 01 D8 0E 1A 11 80:AB
0810 00 CD 05 00 0E 14 11 5C:61
0818 00 CD 05 00 32 75 08 C9:4A
0820 46 69 6C 65 20 6D 6F 64:E0
0828 65 20 30 2E 2E 2E 42 69:EA
0830 6E 20 31 2E 2E 2E 41 73:FD
0838 63 20 00 46 69 6C 65 20:23
0840 6E 61 6D 65 3A 00 53 74:A2
0848 61 72 74 20 41 64 64 72:E2
0850 65 73 73 3A 00 45 78 65:A7
0858 63 20 20 41 64 64 72 65:83
0860 73 73 3A 00 46 69 6C 65:A0
0868 20 20 53 69 7A 65 20 20:1B
0870 20 3A 00 00 00 00 CD 35:5C
0878 02 D8 CD FC 08 D8 01 20:A4
-------------------------------------
SUM: BD BC DC 45 CB 54 8A 8F EB7C

ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum
0880 00 11 F5 0A 2A CF 02 ED:F8
0888 B0 CD BE 0A D8 32 13 0B:6D
0890 32 15 0B AF 32 2B 0B 32:9B
0898 26 0B 21 00 00 22 07 0B:86
08A0 37 3F C9 3A 2B 0B 2A 07:E0
08A8 0B 2C 2D 20 01 3C 67 22:4A
08B0 07 0B 3E 01 ED 5B 27 0B:CB
08B8 21 02 05 CD F3 02 D8 21:E3
08C0 F5 0A ED 5B 29 0B 01 20:9C
08C8 00 ED B0 3E 01 ED 5B 27:4B
08D0 0B 2A D1 02 CD 66 03 D8:16
08D8 06 10 21 15 0B 7E FE 7F:52
08E0 30 12 23 4E E5 21 02 06:C1
08E8 16 00 5F 19 71 E1 05 CA:AF
08F0 F8 08 18 E9 CD AD 0A C9:4E
08F8 3E 07 37 C9 CD 9C 0A D8:90
-------------------------------------
SUM: F4 C8 78 B4 32 19 2F 99 8EBA

ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum
0900 CD 46 09 20 16 7E CD BC:59
0908 09 D8 CD C4 09 D8 E5 01:39
0910 1E 00 09 7E E1 CD 2A 09:86
0918 D8 18 06 CD 81 09 3E 09:94
0920 D8 ED 53 27 0B 22 29 0B:A0
0928 AF C9 D5 E5 11 02 06 6F:BA
0930 26 00 19 7E 36 00 FE 80:71
0938 38 F5 E1 D1 FE 90 30 02:9F
0940 AF C9 3E 07 37 C9 0E 10:DB
0948 ED 5B D5 02 ED 53 27 0B:91
0950 2A D1 02 3E 01 CD F3 02:FE
0958 D8 06 08 22 29 0B 7E FE:B8
0960 FF 28 1A B7 28 0B D5 ED:ED
0968 5B CF 02 CD AD 09 D1 28:A8
0970 0D D5 11 20 00 19 D1 10:0D
0978 E2 13 0D 20 CF 3E AF B7:95
-------------------------------------
SUM: 98 BB 5E B7 C3 3F 43 C2 342A

ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum
0980 C9 C5 0E 10 ED 5B D5 02:CB
0988 21 02 05 3E 01 CD F3 02:29
0990 38 16 06 08 7E B7 28 12:CB
0998 FE FF 28 0E D5 11 20 00:39
09A0 19 D1 10 F0 13 0D 20 E0:0A
09A8 C1 C9 AF C1 C9 C5 E5 06:73
09B0 10 13 23 1A BE 20 02 10:50
09B8 F8 E1 C1 C9 B7 CB 77 C8:24
09C0 3E 04 37 C9 E5 E6 87 21:B5
09C8 96 02 BE E1 C8 3E 06 37:7A
09D0 C9 C5 E5 2A 07 0B 23 22:F4
09D8 07 0B F5 21 02 05 3A FA:63
09E0 09 16 00 5F 19 F1 77 21:20
09E8 FA 09 34 20 0A CD FB 09:32
09F0 3A 26 0B 3C 32 26 0B E1:EB
09F8 C1 C9 00 3A 26 0B 47 3A:76
-------------------------------------
SUM: A4 4E F2 E2 C3 D0 3C 8D 630E

ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum
0A00 2B 0B B8 30 6B 3A 26 0B:F4
0A08 E6 F0 47 3A 2B 0B E6 F0:63
0A10 B8 30 40 CB 3F CB 3F CB:07
0A18 3F CB 3F 21 16 0B 16 00:A1
0A20 5F 19 E5 3A 2B 0B E6 F0:A3
0A28 CD 80 0A CD E1 0A 2A D3:0C
0A30 02 16 00 5F 19 36 8F CD:22
0A38 BE 0A 77 E1 D8 77 2A D3:6C
0A40 02 16 00 5F 19 36 80 3A:80
0A48 2B 0B E6 F0 C6 10 32 2B:3F
0A50 0B 18 A8 3A 26 0B 32 2B:93
0A58 0B CB 3F CB 3F CB 3F CB:F4
0A60 3F 21 16 0B 16 00 5F 19:0F
0A68 3A 26 0B E6 0F C6 80 77:1D
0A70 3A 26 0B CD 80 0A EB 3E:EB
0A78 01 21 02 05 CD 66 03 C9:28
-------------------------------------
SUM: EB 41 DF B4 9E 2F 1A 1B CFE6

ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum
0A80 F5 F5 CB 3F CB 3F CB 3F:08
0A88 CB 3F 21 15 0B 16 00 5F:C0
0A90 19 7E CD D9 0A F1 E6 0F:2D
0A98 85 6F F1 C9 D5 E5 3E 01:A7
0AA0 ED 5B D7 02 21 02 06 CD:17
0AA8 F3 02 E1 D1 C9 D5 E5 3E:68
0AB0 01 ED 5B D7 02 21 02 06:4B
0AB8 CD 66 03 E1 D1 C9 C5 E5:5B
0AC0 06 80 21 02 06 7E B7 28:0C
0AC8 08 23 10 F9 3E 09 37 18:CA
0AD0 05 3E 80 90 37 3F E1 C1:6B
0AD8 C9 26 00 6F 29 29 29 29:02
0AE0 C9 E5 CB 3C CB 1D CB 3C:A4
0AE8 CB 1D CB 3C CB 1D CB 3C:DE
0AF0 CB 1D 7D E1 C9 00 00 00:0F
0AF8 00 00 00 00 00 00 00 00:00
-------------------------------------
SUM: 47 F7 84 D4 75 15 2F 46 9B1E

ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum
0B00 00 00 00 00 00 00 00 00:00
0B08 00 00 00 00 00 00 00 00:00
0B10 00 00 00 00 00 00 00 00:00
0B18 00 00 00 00 00 00 00 00:00
0B20 00 00 00 00 00 00 00 00:00
0B28 00 00 00 00 C9 D5 E5 3E:C1
0B30 01 ED 5B D7 02 21 02 06:4B
0B38 CD 66 03 E1 D1 C9 C5 E5:5B
0B40 06 80 21 02 06 7E B7 28:0C
0B48 08 23 10 F9 3E 09 37 18:CA
0B50 05 3E 80 90 37 3F E1 C1:6B
0B58 C9 26 00 6F 29 29 29 29:02
0B60 C9 E5 CB 3C CB 1D CB 3C:A4
0B68 CB 1D CB 3C CB 1D CB 3C:DE
0B70 CB 1D 7D E1 C9 00 00 00:0F
0B78 00 00 00 00 00 00 00 00:00
-------------------------------------
SUM: 09 79 22 0B 9F E8 3A CB 9C65
```

リスト3 SLOAD

(06FF$_H$まではリスト1と同じ)

```
ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum
0700 00 00 31 00 40 CD 03 01:42
0708 CD 11 07 DC 01 04 C3 00:89
0710 00 11 3A 08 CD FF 01 CD:ED
0718 0E 02 06 01 FE 30 28 02:6F
0720 06 04 78 32 60 08 CD EA:D3
0728 01 11 55 08 CD FF 01 ED:29
0730 5B D9 02 CD 1A 02 CD EA:D6
0738 01 3A 60 08 ED 5B D9 02:C6
0740 CD 62 08 D8 3E FF 32 3E:BC
0748 09 3A 82 00 FE 2D 20 30:40
0750 3A 83 00 FE 54 20 29 3A:92
0758 60 08 FE 04 3E 06 37 C0:A5
0760 CD 1F 09 F5 FE 0D 3E 0A:3D
0768 CC 84 01 F1 CD 84 01 CD:61
0770 71 01 2A B3 09 2B 22 B3:58
0778 09 7C B5 20 E3 37 3F C9:7C
-------------------------------------
SUM: C1 93 18 87 C5 A9 B5 4E 1524

ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum
0780 3E 01 32 5C 00 0E 00 CD:A8
0788 4C 01 7C B5 3E 02 37 C8:BD
0790 0E 13 11 5C 00 CD 05 00:60
0798 0E 16 11 5C 00 CD 05 00:63
07A0 FE FF 3E 04 37 C8 AF 32:1F
07A8 7C 00 CD 1F 09 D8 CD D7:ED
07B0 07 D8 2A B3 09 2B 22 B3:C5
07B8 09 7C B5 20 ED 3A 61 08:EA
07C0 A7 28 04 CD 17 08 D8 0E:A5
07C8 10 11 5C 00 CD 05 00 FE:4D
07D0 FF 3E 04 37 C8 3F C9 F5:3D
07D8 FE 0D 20 13 3A 60 08 FE:DE
07E0 04 20 0C F1 3E 0D CD FE:37
07E8 07 3E 0A CD FE 07 C9 F1:DB
07F0 A7 20 0B 3A 60 08 FE 04:76
07F8 3E 00 20 02 3E 1A F5 3A:E7
-------------------------------------
SUM: D4 80 7F D0 34 91 72 85 BB7B

ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum
0800 61 08 C6 80 26 00 6F F1:35
0808 77 3A 61 08 3C E6 7F 32:ED
0810 61 08 C0 CD 17 08 C9 0E:EC
0818 00 CD 4C 01 7C B5 3E 02:8B
0820 37 C8 0E 1A 11 80 00 CD:85
0828 05 00 0E 15 11 5C 00 CD:62
0830 05 00 FE FF 3E 01 37 C8:40
0838 3F C9 46 69 6C 65 20 6D:15
0840 6F 64 65 20 30 2E 2E 2E:12
0848 42 69 6E 20 31 2E 2E 2E:F4
0850 41 73 63 20 00 46 69 6C:52
0858 65 20 6E 61 6D 65 3A 00:60
0860 00 00 CD 35 02 D8 CD B0:59
0868 08 D8 2A CF 02 11 A1 09:96
0870 01 20 00 ED B0 CD 74 09:08
0878 D8 06 10 0E 00 3A BF 09:FE
-------------------------------------
SUM: F1 06 3E AD 43 DC EC 95 9984

ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum
0880 11 C1 09 12 13 FE 7F 30:AD
0888 0F 2A D3 02 85 6F 30 01:33
0890 24 7E 05 28 17 0C 18 EB:F5
0898 0D F5 79 87 87 87 87 4F:E6
08A0 F1 D6 80 81 32 D7 09 AF:89
08A8 32 D2 09 C9 3E 07 37 C9:1B
08B0 CD C8 08 D8 3E 08 37 C0:B2
08B8 E5 ED 5B CF 02 01 20 00:1F
08C0 ED B0 E1 7E CD 12 09 C9:AD
08C8 0E 10 ED 5B D5 02 ED 53:7D
08D0 D3 09 2A D1 02 3E 01 CD:E5
08D8 F3 02 D8 06 08 22 D5 09:DB
08E0 7E FE FF 28 1A B7 28 0B:A7
08E8 D5 ED 5B CF 02 CD 03 09:C7
08F0 D1 28 0D D5 11 20 00 19:25
08F8 D1 10 E2 13 0D 20 CF 3E:10
-------------------------------------
SUM: DC A9 5F 43 CC 1F AB 00 6D2D

ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum
0900 AF B7 C9 C5 E5 06 10 13:02
0908 23 1A BE 20 02 10 F8 E1:06
0910 C1 C9 E5 E6 87 21 96 02:95
0918 BE E1 C8 3E 06 37 C9 21:CC
0920 3E 09 34 20 0B CD 3F 09:BB
0928 D8 3A D2 09 3C 32 D2 09:36
0930 21 02 05 3A 3E 09 16 00:BF
0938 5F 19 7E 37 3F C9 00 3A:6F
0940 D2 09 47 3A D7 09 B8 DA:CE
0948 AC 08 78 CD 58 09 EB 3E:83
0950 01 21 02 05 CD F3 02 C9:B4
0958 F5 F5 CB 3F CB 3F CB 3F:08
0960 CB 3F 21 C1 09 16 00 5F:6A
0968 19 7E CD 85 09 F1 E6 0F:D8
0970 85 6F F1 C9 D5 E5 3E 01:A7
0978 ED 5B D7 02 2A D3 02 CD:ED
-------------------------------------
SUM: B1 87 FF FF 10 42 24 BF 2E52

ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum
0980 F3 02 E1 D1 C9 26 00 6F:05
0988 29 29 29 29 C9 E5 CB 3C:59
0990 CB 1D CB 3C CB 1D CB 3C:DE
0998 CB 1D CB 3C CB 1D 7D E1:35
09A0 C9 00 00 00 00 00 00 00:C9
09A8 00 00 00 00 00 00 00 00:00
09B0 00 00 00 00 00 00 00 00:00
09B8 00 00 00 00 00 00 00 00:00
09C0 00 00 00 00 00 00 00 00:00
09C8 00 00 00 00 00 00 00 00:00
09D0 00 00 00 00 00 00 00 00:00
09D8 F5 F5 CB 3F CB 3F CB 3F:08
09E0 CB 3F 21 C1 09 16 00 5F:6A
09E8 19 7E CD 85 09 F1 E6 0F:D8
09F0 85 6F F1 C9 D5 E5 3E 01:A7
09F8 ED 5B D7 02 2A D3 02 CD:ED
-------------------------------------
SUM: C6 E1 21 C2 04 43 04 43 3262
```

▶マシン語もわからないまま，マシン語で画像圧縮ツールを作ろうとしている自分が恐ろしい。
中島　健喜（16）大阪府

リスト4 CRC

```
ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum
0100 31 00 40 0E 0F 11 5C 00:FB
0108 CD 05 00 FE FF 28 17 21:2F
0110 00 01 22 93 02 0E 14 11:EB
0118 5C 00 CD 05 00 A7 C2 00:97
0120 00 CD 40 01 18 EF 11 2F:55
0128 01 CD 45 02 C3 00 00 46:1E
0130 69 6C 65 20 6E 6F 74 20:CB
0138 66 6F 75 6E 64 0D 0A 00:33
0140 11 68 02 CD 45 02 21 80:30
0148 00 0E 10 CD EE 01 CD 62:09
0150 01 CD 30 02 11 08 00 19:32
0158 0D 20 F0 CD 8F 01 CD 30:77
0160 02 C9 E5 2A 93 02 CD 27:63
0168 02 11 08 00 19 22 93 02:EB
0170 E1 E5 06 08 1E 00 CD 3D:FC
0178 02 7E 23 F5 CD 10 02 F1:68
--------------------------------------
SUM: 30 1B D6 C5 27 99 C2 49 8BB2

ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum
0180 83 5F 10 F2 3E 3A CD 01:2A
0188 02 7B CD 10 02 E1 C9 06:0C
0190 21 3E 2D CD 01 02 10 FB:67
0198 CD 30 02 11 8D 02 CD 45:B1
01A0 02 21 80 00 0E 08 E5 06:A4
01A8 10 AF 86 11 08 00 19 10:87
01B0 F9 CD 10 02 CD 3D 02 E1:C5
01B8 23 0D 20 EA 21 80 00 06:E1
01C0 7E CD C8 01 CD 27 02 C9:D3
01C8 56 23 5E 23 D5 1E 80 D9:46
01D0 E1 D9 7E A3 28 01 37 D9:14
01D8 ED 6A 30 08 3E 10 AC 67:F0
01E0 3E 21 AD 6F D9 CB 0B 30:5A
01E8 E9 23 10 E6 D9 C9 E5 D5:5E
01F0 C5 0E 06 1E FF CD 05 00:C8
01F8 FE 03 CA 00 00 C1 D1 E1:3E
--------------------------------------
SUM: 2D 7A A3 1F 8B 5C 9E 0C 8D40

ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum
0200 C9 E5 D5 C5 F5 0E 02 5F:AC
0208 CD 05 00 F1 C1 D1 E1 C9:FF
0210 F5 0F 0F 0F 0F CD 19 02:19
0218 F1 E6 0F C6 30 FE 3A 38:4C
0220 02 C6 07 CD 01 02 C9 7C:E4
0228 CD 10 02 7D CD 10 02 C9:04
0230 F5 3E 0D CD 01 02 3E 0A:58
0238 CD 01 02 F1 C9 F5 3E 20:DD
0240 CD 01 02 F1 C9 D5 F5 1A:6E
0248 A7 28 06 CD 01 02 13 18:D0
0250 F6 F1 D1 C9 49 4E 50 55:BD
0258 54 20 45 4E 44 20 41 44:F0
0260 44 52 45 53 53 20 3A 00:DB
0268 0D 0A 41 44 52 53 20 2B:8C
0270 30 20 2B 31 20 2B 32 20:49
0278 2B 33 20 2B 34 20 2B 35:5D
--------------------------------------
SUM: 77 DD FA 5B DD B6 CD 1C 8912

ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum
0280 20 2B 36 20 2B 37 20 53:76
0288 75 6D 0D 0A 00 53 55 4D:EE
0290 3A 20 00 00 00 CD 19 02:42
0298 F1 E6 0F C6 30 FE 3A 38:4C
02A0 02 C6 07 CD 01 02 C9 7C:E4
02A8 CD 10 02 7D CD 10 02 C9:04
02B0 F5 3E 0D CD 01 02 3E 0A:58
02B8 CD 01 02 F1 C9 F5 3E 20:DD
02C0 CD 01 02 F1 C9 D5 F5 1A:6E
02C8 A7 28 06 CD 01 02 13 18:D0
02D0 F6 F1 D1 C9 49 4E 50 55:BD
02D8 54 20 45 4E 44 20 41 44:F0
02E0 44 52 45 53 53 20 3A 00:DB
02E8 0D 0A 41 44 52 53 20 2B:8C
02F0 30 20 2B 31 20 2B 32 20:49
02F8 2B 33 20 2B 34 20 2B 35:5D
--------------------------------------
SUM: BB 9C 59 C0 43 61 5F 94 4CD5
```

リスト5 SDIRソースリスト

```
==================  sdir.mac  ==================
   1: ;**********************************
   2: ;
   3: ; S-OS 'Sword' Dir command Program
   4: ;
   5: ;        Programed by T.Ishigami
   6: ;
   7: ;          '87 Nov.12th
   8: ;
   9: ;**********************************
  10:
  11:         .Z80
  12:
  13:         aseg
  14:         org     103h
  15:
  16: INCLUDE SOS.SIM
  17:
  18: start:  ld      sp,4000h
  19:         call    cpminit
  20:         call    dir
  21:         call    c,error
  22:         jp      0
  23:
  24:
  25: ;====================
  26: ; Directory Display
  27: ;====================
  28:
  29: dir:    call    fatred
  30:         ret     c       ;Disk error
  31:
  32:         ld      a,'$'
  33:         call    _print
  34:         call    freclu
  35:         call    _prthx
  36:         ld      de,cstmes
  37:         call    _msx
  38:         ld      b,16
  39:         ld      de,(_dirps)     ;Directory start
  40: dirl:   ld      hl,dmabf
  41:         ld      a,1
  42:         call    _drdsb
  43:         ret     c       ;Disk error
  44:         call    dirprt
  45:         jr      z,dirs
  46:         inc     de
  47:         djnz    dirl
  48: dirs:   ret
  49:
  50:
  51: dirprt: push    bc
  52:         push    de
  53:         ld      b,8
  54: dirpl:  ld      a,(hl)
  55:         and     a
  56:         jr      z,dirn
  57:         cp      0ffh
  58:         jr      z,dirpe
  59:         call    p_fnam
  60:         call    _ltnl
  61:         call    _pause
  62: dirn:   ld      de,20H
  63:         add     hl,de
  64:         djnz    dirpl
  65:         db      3eh     ;z = 0
  66: dirpe:  xor     a       ;z = 1
  67:         pop     de
  68:         pop     bc
  69:         or      a
  70:         ret
  71:
  72:
  73: p_fnam: push    bc
  74:         push    de
  75:         push    hl
  76:         ld      de,_ibf
  77:         ld      bc,20h
  78:         ldir
  79:         call    atrprt
  80:         call    _fprnt
  81:
  82:
  83: ; put stadr, edadr, exadr
  84:
  85:         ld      hl,(_ibf+14h)   ;_stadr
  86:         call    phex
  87:
  88:         ld      de,(_ibf+12h)   ;_size
  89:         add     hl,de
  90:         dec     hl
  91:         call    phex
  92:
  93:         ld      hl,(_ibf+16h)   ;_exadr
  94:         call    phex
  95:
  96:         pop     hl
  97:         pop     de
  98:         pop     bc
  99:         ret
 100:
 101: phex:   ld      a,':'
 102:         call    _print
 103:         call    _prthl
 104:         ret
 105:
 106:
 107: ; File Attribute print
 108:
 109: atrprt: ld      a,(_ibf)
 110:         bit     7,a
 111:         jr      z,atrpl
 112:         ld      a,8
 113:         db      11h     ;skip next opration
 114: atrpl:  and     7
 115:         ld      l,a
 116:         ld      h,0
 117:         add     hl,hl
 118:         add     hl,hl
 119:         ld      de,atrmes
 120:         add     hl,de
 121:         ex      de,hl
 122:         call    _msx
 123:
 124:         ld      a,(_ibf)
 125:         bit     6,a
 126:         ld      a,'*'
 127:         jr      nz,atrp2
 128:         ld      a,' '
 129: atrp2:  call    _print
 130:         call    _prnts
 131:         ret
 132:
 133: _fprnt: ld      de,_ibf+1
 134:         ld      b,13
 135:         ld      a,(de)
 136:         cp      ' '
 137:         jr      nc,fprnt1
 138:         ld      a,' '
 139: fprnt1: cp      '.'
 140:         jr      nz,fprnt2
 141:         ld      a,' '
 142: fprnt2: call    _print
 143:         inc     de
 144:         djnz    _fprnt+5
 145:
 146:
 147: ; ext print
 148:
 149:         ld      a,'.'
 150:         call    _print
 151:         ld      b,3
 152: filpr2: ld      a,(de)
 153:         cp      ' '
 154:         jr      nc,filpr3
 155:         ld      a,' '
 156: filpr3: call    _print
 157:         inc     de
 158:         djnz    filpr2
 159:         ret
 160:
 161:
 162: freclu: push    bc
 163:         push    hl
 164:         ld      b,80h
 165:         ld      c,0
 166:         ld      hl,(_fatbf)
 167: frecl1: ld      a,(hl)
 168:         and     a
 169:         jr      nz,frecl2
 170:         inc     c
 171: frecl2: inc     hl
 172:         djnz    frecl1
 173:         ld      a,c
 174:         pop     hl
 175:         pop     bc
 176:         ret
 177:
 178:
 179: fatred: push    de
 180:         push    hl
 181:         ld      de,(_fatpos)
 182:         ld      hl,(_fatbf)
 183:         ld      a,1
 184:         call    _drdsb
 185:         pop     hl
 186:         pop     de
 187:         ret
 188:
 189:
 190: atrmes: db      'Nul',0 ;0
 191:         db      'Bin',0 ;1
 192:         db      'Bas',0 ;2
 193:         db      '???',0 ;3
 194:         db      'Asc',0 ;4
 195:         db      '???',0 ;5
 196:         db      '???',0 ;6
 197:         db      '???',0 ;7
 198:         db      'Dir',0 ;<=80h
 199:
 200:
 201: cstmes: db      ' Clusters Free',0dh,0ah,0
 202:
 203:
 204: _ibf:   ds      20h
 205:
 206:         end     start
```

リスト6 SSAVEソースリスト

```
==================  ssave.mac  ==================
   1: ;**********************************
   2: ;
   3: ; S-OS 'Sword' Save command Program
   4: ;
   5: ;        Programed by T.Ishigami
   6: ;
   7: ;          '87 Nov.18th
   8: ;
   9: ;**********************************
  10:
  11:         .z80
  12:
  13:         aseg
  14:         org     103h
  15:
  16: include SOS.SIM
  17:
  18:
  19: start:  ld      sp,4000h
  20:         call    cpminit
  21:         call    main
  22:         call    c,error
  23:         jp      0
  24:
  25: main:   ld      de,msg1
  26:         call    _msx
  27:         call    _flget
  28:         ld      b,1     ;Bin
  29:         cp      '0'             ;0 = Bin ,other = Asc
  30:         jr      z,skip1
  31:         ld      b,4     ;Asc
  32: skip1:  ld      a,b
  33:         ld      (flmode),a
  34:         call    _ltnl
  35:
  36:         ld      de,msg2
  37:         call    _msx
  38:         ld      de,(_kbfad)
  39:         call    _getl
  40:         call    _ltnl
  41:
  42:         xor     a
  43:         ld      (wrpnt),A
  44:         ld      (cpmpnt),a
  45:
  46:         ld      a,1     ;Select A drive
  47:         ld      (5ch),a
  48:         ld      c,0
  49:         call    seldsk
  50:         ret     c
  51:
  52:         ld      c,15    ;Open File
  53:         ld      de,5ch
  54:         call    bdos
  55:         cp      0ffh
  56:         ld      a,8     ;File not found
  57:         scf
  58:         ret     z
  59:
  60:         ld      a,(flmode)
  61:         ld      de,(_kbfad)
  62:         call    wropen
  63:         ret     c
  64:
  65:         xor     a       ;clear 'cr' field
  66:         ld      (5ch+32),a
  67:
  68:         ld      a,(flmode)
  69:         cp      4
  70:         jr      z,ascfile
  71:
  72: binfile:ld      de,msg3
  73:         call    _msx
  74:         ld      de,(_kbfad)
  75:         call    _getl
  76:         call    _ltnl
  77:         call    _hlhex
  78:         ld      (FLDTADR),hl
  79:         ret     c
  80:
  81:         ld      de,msg4
  82:         call    _msx
  83:         ld      de,(_kbfad)
  84:         call    _getl
  85:         call    _ltnl
  86:         call    _hlhex
  87:         ld      (FLEXADR),hl
  88:         ret     c
  89:
  90: loop1:  call    cpmget
  91:         call    print_
  92:         ret     c
  93:
  94:         ld      a,(endflg)
  95:         and     a
  96:         jr      z,loop1
  97:         jr      skip5
  98:
  99: ascfile:call    cpmget
 100:         cp      0ah
 101:         jr      z,ascfile
 102:         cp      1ah     ;EOF
 103:         jr      z,skip4
 104:         call    print_
 105:         ld      a,(endflg)
```


▶ 日曜日夜7時からやっているアニメ,「キテレツ大百科」に出てくる主人公が使っているパソコンはなんとX1turbo Z IIだった。

浪越　考宅（16）和歌山県

```
106:         and     a
107:         jr      z,ascfile
108: skip4:  xor     a       ;EOF
109:         call    print_
110:
111: skip5:  ld      de,msg5
112:         call    _msx
113:         ld      hl,(flsize)
114:         call    _prthl
115:         inc     l
116:         dec     l
117:         call    nz,fwrite
118:         call    close
119:         ret
120:
121: cpmget: xor     a
122:         ld      (endflg),a
123:
124:         ld      a,(cpmpnt)
125:         and     7fh
126:         push    af
127:         call    z,get1
128:         pop     af
129:         add     a,80h
130:         ld      h,0
131:         ld      l,a
132:         ld      a,(hl)
133:
134:         push    af
135:         ld      a,(cpmpnt)
136:         inc     a
137:         and     7fh
138:         ld      (cpmpnt),a
139:         jr      nz,skip2
140:         ld      hl,endflg
141:         ld      a,(hl)
142:         ld      (hl),0  ;non end
143:         and     a
144:         jr      z,skip2
145:         ld      (hl),1  ;end
146: skip2:  pop     af
147:         ret
148:
149: get1:   ld      c,0
150:         call    seldsk
151:         ret     c
152:
153:         ld      c,26
154:         ld      de,80h
155:         call    bdos    ;set dma
156:
157:         ld      c,20    ;Sequential read
158:         ld      de,5ch
159:         call    bdos
160:         ld      (endflg),a
161:         ret
162:
163:
164: msg1:   db      'File mode 0...Bin 1...Asc ',0
165: msg2:   db      'File name:',0
166: msg3:   db      'Start Address:',0
167: msg4:   db      'Exec  Address:',0
168: msg5:   db      'File  Size   :',0
169:
170: flmode: ds      1
171: cpmpnt: ds      1
172: endflg: ds      1
173:
174:
175: ;====================
176: ; FILE OPEN FOR WRITE
177: ;====================
178:
179: WROPEN: CALL    _FILE
180:         RET     C
181:         CALL    _WOPEN
182:         RET     C
183:         ;
184:         LD      BC,20H
185:         LD      DE,FILE_BF
186:         LD      HL,(_IBFAD)
187:         LDIR
188:
189:         ;CALL   RDFAT               ;FAT has already loaded.
190:
191:         CALL    FCGET
192:         RET     C
193:
194:         LD      (FSTCLST),A
195:         LD      (TBLCLST),A
196:
197:         XOR     A
198:         LD      (RC),A
199:         LD      (FLPNT),A
200:
201:         LD      HL,0
202:         LD      (FLSIZE),HL
203:
204:         RCF
205:         RET
206:
207:
208: ;=============
209: ; FILE CLOSE
210: ;=============
211:
212: CLOSE:  LD      A,(RC)
213:         LD      HL,(FLSIZE)
214:         INC     L
215:         DEC     L
216:         JR      NZ,COL3
217:         INC     A
218: COL3:   LD      H,A
219:         LD      (FLSIZE),HL
220:
221:         LD      A,1
222:         LD      DE,(DEBUF)
223:         LD      HL,DMABF
224:         CALL    _DRDSB
225:         RET     C
226:
227:         LD      HL,FILE_BF
228:         LD      DE,(HLBUF)
229:         LD      BC,20H
230:         LDIR
231:
232:         LD      A,1
233:         LD      DE,(DEBUF)
234:         LD      HL,(_DTBUF)
235:         CALL    _DWTSB
236:         RET     C
237:
238:         LD      B,10H
239:         LD      HL,TBLCLST
240: COL1:   LD      A,(HL)
241:         CP      7FH
242:         JR      NC,COL2
243:
244:         INC     HL
245:         LD      C,(HL)
246:         PUSH    HL
247:         LD      HL,FATBF
248:         LD      D,0
249:         LD      E,A
250:         ADD     HL,DE
251:         LD      (HL),C
252:         POP     HL
253:
254:         DEC     B
255:         JP      Z,RDOPN3            ;Bad File Allocation
256:         JR      COL1
257:
258: COL2:   CALL    WRFAT
259:         RET
260:
261: RDOPN3: LD      A,7
262:         SCF
263:         RET
264:
265: _wopen: call    rdfat
266:         ret     c       ;Disk error
267:         call    fcbsch
268:         jr      nz,wopen2
269:
270:         ld      a,(hl)
271:         call    wpchk
272:         ret     c       ;Write protected
273:         call    fmchk
274:         ret     c       ;Bad file mode
275:
276:         push    hl
277:         ld      bc,1eh
278:         add     hl,bc
279:         ld      a,(hl)
280:         pop     hl
281:         call    erafat
282:         ret     c       ;Bad allocation table
283:         jr      wopen3
284:
285: wopen2: call    fresch
286:         ld      a,9     ;Device Full
287:         ret     c
288: wopen3: ld      (debuf),de
289:         ld      (hlbuf),hl
290:         xor     a
291:         ret
292:
293: ; Fat Erase
294:
295: erafat: push    de
296:         push    hl
297:         ld      de,fatbf
298: erafa1: ld      l,a
299:         ld      h,0
300:         add     hl,de
301:         ld      a,(hl)
302:         ld      (hl),0
303:         cp      80h
304:         jr      c,erafa1
305:         pop     hl
306:         pop     de
307:         cp      90h
308:         jr      nc,erafa2
309:         xor     a
310:         ret
311:
312: erafa2: ld      a,7     ;Bad Allocation table
313:         scf
314:         ret
315:
316:
317: ;
318: ; FCB SEARCH
319: ;
320:
321: FCBSCH: LD      C,16            ;Directory Lengh
322:         LD      DE,(_DIRPS)     ;Directory start
323:
324: FCBSC1: LD      (DEBUF),DE
325:         LD      HL,(_DTBUF)
326:         LD      A,1
327:         CALL    _DRDSB
328:         RET     C
329:         LD      B,8
330:
331: FCBSC2: LD      (HLBUF),HL
332:         LD      A,(HL)
333:         CP      0FFH
334:         JR      Z,FCBSC4
335:         OR      A
336:         JR      Z,FCBSC3
337:         PUSH    DE
338:         LD      DE,(_IBFAD)
339:         CALL    FCOMP
340:         POP     DE
341:         JR      Z,FCBSC5
342:
343: FCBSC3: PUSH    DE
344:         LD      DE,32
345:         ADD     HL,DE
346:         POP     DE
347:         DJNZ    FCBSC2
348:         INC     DE
349:         DEC     C
350:         JR      NZ,FCBSC1
351:
352: FCBSC4: DB      3EH     ; Z = 0
353: FCBSC5: XOR     A       ; Z = 1
354:         OR      A
355:         RET
356:
357: ; free fcb search
358: fresch: push    bc
359:         ld      c,16            ;Directory lengh
360:         ld      de,(_dirps)     ;Directory start
361: fresc1: ld      hl,dmabf
362:         ld      a,1
363:         call    _drdsb
364:         jr      c,fresc3
365:         ld      b,8
366: fresc2: ld      a,(hl)
367:         or      a
368:         jr      z,fresc4
369:         cp      0ffh
370:         jr      z,fresc4
371:         push    de
372:         ld      de,32
373:         add     hl,de
374:         pop     de
375:         djnz    fresc2
376:         inc     de
377:         dec     c
378:         jr      nz,fresc1
379:
380: fresc3: pop     bc
381:         ret
382:
383: fresc4: xor     a
384:         pop     bc
385:         ret
386:
387:
388: ; File Name Compare
389: FCOMP:  PUSH    BC
390:         PUSH    HL
391:         LD      B,16    ;Directory lengh
392: FCOMP1: INC     DE
393:         INC     HL
394:         LD      A,(DE)
395:         CP      (HL)
396:         JR      NZ,FCOMP2
397:         DJNZ    FCOMP1
398:
399: FCOMP2: POP     HL
400:         POP     BC
401:         RET
402:
403:
404: ; file write protected check
405:
406: wpchk:  or      a
407:         bit     6,a
408:         ret     z
409:         ld      a,4     ;Write protected
410:         scf
411:         ret
412:
413:
414: ; file mode check
415:
416: fmchk:  push    hl
417:         and     87h     ;1000 0111b
418:         ld      hl,@ibuf
419:         cp      (hl)
420:         pop     hl
421:         ret     z
422:         ld      a,6     ;Bad file mode
423:         scf
424:         ret
425:
426:
427: ;===============
428: ; PRINT TO FILE
429: ;===============
430:
431: PRINT_: PUSH    BC
432:         PUSH    HL
433:
434:         LD      HL,(FLSIZE)
435:         INC     HL
436:         LD      (FLSIZE),HL
437:
438:         PUSH    AF
439:         LD      HL,DMABF
440:         LD      A,(WRPNT)
441:         LD      D,0
442:         LD      E,A
443:         ADD     HL,DE
444:         POP     AF
445:         LD      (HL),A
446:
447:         LD      HL,WRPNT
448:         INC     (HL)
449:         JR      NZ,PR1
450:
451:         CALL    FWRITE
452:         LD      A,(FLPNT)
453:         INC     A
454:         LD      (FLPNT),A
455:
456: PR1:    POP     HL
457:         POP     BC
458:         RET
459:
460: WRPNT:  DS      1
461:
462:
463: FWRITE: LD      A,(FLPNT)
464:         LD      B,A
465:         LD      A,(RC)
466:         CP      B
467:         JR      NC,FWRITE4
468:
469:         LD      A,(FLPNT)
470:         AND     0F0H
471:         LD      B,A
472:         LD      A,(RC)
473:         AND     0F0H
474:         CP      B
475:         JR      NC,FWRITE2
476:
477:         SRL     A
478:         SRL     A
479:         SRL     A
480:         SRL     A               ;A = A / 16
481:         LD      HL,TBLCLST+1
482:         LD      D,0
483:         LD      E,A
484:         ADD     HL,DE
485:
486:         PUSH    HL
487:         LD      A,(RC)
488:         AND     0F0H
489:         CALL    PNTREC
490:         CALL    RECCL
491:
492:         LD      HL,(_FATBF)
493:         LD      D,0
494:         LD      E,A             ;A = RC && 0F0H
495:         ADD     HL,DE
496:         LD      (HL),8FH        ;DUMMY
497:         CALL    FCGET
498:         LD      (HL),A
499:         POP     HL
500:         RET     C
501:         LD      (HL),A
502:
503:         LD      HL,(_FATBF)
504:         LD      D,0
505:         LD      E,A
506:         ADD     HL,DE
507:         LD      (HL),80H
508:         LD      A,(RC)
509:         AND     0F0H
510:         ADD     A,16            ;ONE CLUSTER ADDED
511:         LD      (RC),A
512:         JR      FWRITE
513:
514: FWRITE2:LD      A,(FLPNT)
515:         LD      (RC),A
516:         srl     a
517:         srl     a
518:         srl     a
519:         srl     a       ;a = a / 16
520:         ld      hl,tblclst+1
521:         ld      d,0
522:         ld      e,a
523:         add     hl,de
524:
525:         ld      a,(flpnt)
526:         and     0fh
527:         add     a,80h
528:         ld      (hl),a
529:
530: FWRITE4:LD      A,(FLPNT)
531:         CALL    PNTREC
532:         EX      DE,HL           ;DE = RECORD NO.
533:
534:         LD      A,1
535:         LD      HL,DMABF
536:         CALL    _DWTSB
537:         RET
538:
539:
540: ; FILE POINTER (A) => RECORD NO. (HL)
541:
542: PNTREC: PUSH    AF
543:         PUSH    AF
544:         SRL     A
545:         SRL     A
546:         SRL     A
547:         SRL     A               ;A = A / 16
548:         LD      HL,TBLCLST
549:         LD      D,0
550:         LD      E,A
551:         ADD     HL,DE
552:         LD      A,(HL)
553:         CALL    CLREC
554:         POP     AF
555:         AND     0FH
556:         ADD     A,L
557:         LD      L,A
558:         POP     AF
559:         RET
560:
561:
562: ; FAT READ TO BUFFER
563:
564: RDFAT:  PUSH    DE
565:         PUSH    HL
566:         LD      A,1
567:         LD      DE,(_FATPOS)
568:         LD      HL,FATBF
569:         CALL    _DRDSB
570:         POP     HL
571:         POP     DE
572:         RET
573:
```

▶隠された真実シリーズ。X68000とPC-286のネーミングのコンセプトは一緒だ。
小松　恭郎（17）長野県

```
574:
575: ; FAT WRITE FROM BUFFER
576:
577: WRFAT:  PUSH    DE
578:         PUSH    HL
579:         LD      A,1
580:         LD      DE,(_FATPOS)
581:         LD      HL,FATBF
582:         CALL    _DWTSB
583:         POP     HL
584:         POP     DE
585:         RET
586:
587: ; FREE CLUSTER POSITION GET
588:
589: FCGET:  PUSH    BC
590:         PUSH    HL
591:         LD      B,80H
592:         LD      HL,FATBF
593: FCGET2: LD      A,(HL)
594:         OR      A
595:         JR      Z,FCGET3
596:         INC     HL
597:         DJNZ    FCGET2
598:         LD      A,9             ;Device Full
599:         SCF
600:         JR      FCGET4
601: FCGET3: LD      A,80H
602:         SUB     B
603:         RCF
604: FCGET4: POP     HL
605:         POP     BC
606:         RET
607:
608: ; CLUSTER (A) => RECORD (HL)
609:
610: CLREC:  LD      H,0
611:         LD      L,A
612:         ADD     HL,HL
613:         ADD     HL,HL
614:         ADD     HL,HL
615:         ADD     HL,HL
616:         RET
617:
618: ; RECORD (HL) => CLUSTER (A)
619:
620: RECCL:  PUSH    HL
621:         SRL     H
622:         RR      L       ;HL/2
623:         SRL     H
624:         RR      L       ;HL/4
625:         SRL     H
626:         RR      L       ;HL/8
627:         SRL     H
628:         RR      L       ;HL/16
629:         LD      A,L
630:         POP     HL
631:         RET
632:
633:
634: IMF_SIZE  EQU  37H
635: FILE_BF:DS  IMF_SIZE
636: FLSIZE  EQU  FILE_BF+12H  ;DS 2  ;File Size.
637: FLDTADR  EQU  FILE_BF+14H  ;DS 2  ;Start Address.
638: FLEXADR  EQU  FILE_BF+16H  ;DS 2  ;Exec Address.
639: FSTCLST  EQU  FILE_BF+1EH  ;DS 1  ;First Cluster.
640: TBLCLST  EQU  FILE_BF+20H  ;DS 20H ;Cluster table.
641: FLPNT  EQU  FILE_BF+31H  ;DS 1  ;The FILE Pointor.
642: DEBUF  EQU  FILE_BF+32H  ;DS 2  ;Record  No. Which Have The DIR.
643: HLBUF  EQU  FILE_BF+34H  ;DS 2  ;Address Where On IMFORMATION.
644: RC  EQU  FILE_BF+36H  ;DS 1  ;The Number of Records The File  have.
645:
646:   END  start
```

リスト7 SLOADソースリスト

```
==================  sload.mac  ==================
    1: ;**********************************
    2: ; S-OS Load Command  Program
    3: ;        Programed by T.Ishigami
    4: ;          '87 Nov.14th
    5: ;
    6: ;**********************************
    7:         .Z80
    8:
    9:         ASEG
   10:         ORG     103H
   11:
   12: INCLUDE SOS.SIM
   13:
   14:
   15: start:  ld      sp,4000h
   16:         call    cpminit
   17:         call    main
   18:         call    c,error
   19:         jp      0
   20:
   21:
   22: main:   ld      de,msg1
   23:         call    _msx
   24:         call    _flget
   25:         ld      b,1     ;Bin
   26:         cp      '0'             ;0 = Bin ,other = Asc
   27:         jr      z,skip1
   28:         ld      b,4     ;Asc
   29: skip1:  ld      a,b
   30:         ld      (flmode),a
   31:         call    _ltnl
   32:
   33:         ld      de,msg2
   34:         call    _msx
   35:         ld      de,(_kbfad)
   36:         call    _getl
   37:         call    _ltnl
   38:
   39:         LD      A,(flmode)
   40:         LD      DE,(_kbfad)
   41:         CALL    RDOPEN
   42:         RET     C
   43:
   44:         LD      A,-1
   45:         LD      (RDPNT),A
   46:
   47:         ld      a,(80h+2)
   48:         cp      '-'
   49:         jr      nz,to_file
   50:         ld      a,(80h+3)
   51:         cp      'T'
   52:         jr      nz,to_file
   53:
   54: ; type out.
   55: ;
   56: to_crt: ld      a,(flmode)
   57:         cp      4
   58:         ld      a,6     ;Bad File mode
   59:         scf
   60:         ret     nz
   61:
   62: loop:   call    input_
   63:         push    af
   64:         cp      0dh
   65:         ld      a,0ah
   66:         call    z,_print
   67:         pop     af
   68:         call    _print
   69:
   70:         call    _pause
   71:
   72:         ld      hl,(flsize)
   73:         dec     hl
   74:         ld      (flsize),hl
   75:         ld      a,h
   76:         or      l
   77:         JR      NZ,LOOP
   78:         rcf
   79:         RET
   80:
   81:
   82: to_file:ld      a,1     ;Select A drive
   83:         ld      (5ch),a
   84:         ld      c,0
   85:         call    seldsk
   86:         ld      a,h
   87:         or      l
   88:         ld      a,2     ;Device offline
   89:         scf
   90:         ret     z
   91:
   92:         ld      c,19    ;Delete File if it exists
   93:         ld      de,5ch
   94:         call    bdos
   95:
   96:         ld      c,22    ;Make File
   97:         ld      de,5ch
   98:         call    bdos
   99:         cp      0ffh
  100:         ld      a,4     ;Write Proctcted
  101:         scf
  102:         ret     z
  103:
  104:         xor     a       ;clear 'cr' field
  105:         ld      (5ch+32),a
  106:
  107:
  108: loop3:  call    input_
  109:         ret     c
  110:         call    cpmput
  111:         ret     c
  112:         ld      hl,(flsize)
  113:         dec     hl
  114:         ld      (flsize),hl
  115:         ld      a,h
  116:         or      l
  117:         jr      nz,loop3
  118:
  119:         ld      a,(cpmpnt)
  120:         and     a
  121:         jr      z,skip3
  122:         call    put3
  123:         ret     c
  124:
  125: skip3:  ld      c,16
  126:         ld      de,5ch
  127:         call    bdos    ;close file
  128:         cp      0ffh
  129:         ld      a,4
  130:         scf
  131:         ret     z
  132:         ccf
  133:         ret
  134:
  135: cpmput: push    af
  136:         cp      0dh
  137:         jr      nz,cpm1
  138:
  139:         ld      a,(flmode)
  140:         cp      4       ;Asc ?
  141:         jr      nz,cpm1
  142:
  143:         pop     af
  144:         ld      a,0dh
  145:         call    cpm2
  146:         ld      a,0ah
  147:         call    cpm2
  148:         ret
  149:
  150: cpm1:   pop     af
  151:         and     a
  152:         jr      nz,cpm2         ;convert eof code
  153:         ld      a,(flmode)      ;00(S-OS) => 1A(CP/M)
  154:         cp      4       ;Asc ?
  155:         ld      a,0
  156:         jr      nz,cpm2
  157:         ld      a,1ah   ;eof
  158:
  159: cpm2:   push    af
  160:         ld      a,(cpmpnt)
  161:         add     a,80h
  162:         ld      h,0
  163:         ld      l,a
  164:         pop     af
  165:         ld      (hl),a
  166:         ld      a,(cpmpnt)
  167:         inc     a
  168:         and     7fh
  169:         ld      (cpmpnt),a
  170:         ret     nz
  171:         call    put3
  172:         ret
  173:
  174:
  175: put3:   ld      c,0
  176:         call    seldsk
  177:         ld      a,h
  178:         or      l
  179:         ld      a,2     ;Device offline
  180:         scf
  181:         ret     z
  182:
  183:         ld      c,26
  184:         ld      de,80h
  185:         call    bdos    ;set dma
  186:
  187:         ld      c,21
  188:         ld      de,5ch
  189:         call    bdos
  190:         cp      0ffh
  191:         ld      a,1     ;Device I/O Error
  192:         scf
  193:         ret     z
  194:         ccf
  195:         ret
  196:
  197:
  198: msg1:   db      'File mode 0...Bin 1...Asc ',0
  199: msg2:   db      'File name:',0
  200: flmode: ds      1
  201: cpmpnt: ds      1
  202:
  203: ;====================
  204: ; FILE OPEN FOR READ
  205: ;====================
  206: RDOPEN: CALL    _FILE
  207:         RET     C
  208:         CALL    _ROPEN
  209:         RET     C
  210:         ;
  211:         LD      HL,(_IBFAD)
  212:         LD      DE,FILE_BF
  213:         LD      BC,20H
  214:         LDIR
  215:
  216:         CALL    RDFAT
  217:         RET     C
  218:
  219:         LD      B,10H
  220:         LD      C,0             ; C <= TOTAL NUMBER OF CLUSTERS
  221:         LD      A,(FSTCLST)
  222:         LD      DE,TBLCLST
  223:
  224: RDOPN4: LD      (DE),A
  225:         INC     DE
  226:         CP      7FH
  227:         JR      NC,RDOPN5
  228:         LD      HL,(_FATBF)
  229:         ADD     A,L
  230:         LD      L,A
  231:         JR      NC,RDOPN2
  232:         INC     H
  233: RDOPN2: LD      A,(HL)          ;HL = (_FATBF) + (FSTCLST)
  234:         DEC     B
  235:         JR      Z,RDOPN3        ; MORE THAN 16 CLUSTER
  236:         INC     C
  237:         JR      RDOPN4
  238:
  239: RDOPN5: DEC     C               ;LAST CLUSTER IS DUMMY
  240:         PUSH    AF
  241:         LD      A,C
  242:         ADD     A,A
  243:         ADD     A,A
  244:         ADD     A,A
  245:         ADD     A,A
  246:         LD      C,A     ;C = C * 16
  247:         POP     AF
  248:         SUB     80H     ;RC is 0 origin. So it isn't 7Fh but 80H.
  249:         ADD     A,C
  250:         LD      (RC),A  ;RC = Total number of Records
  251:
  252:         XOR     A
  253:         LD      (FLPNT),A
  254:         RET
  255:
  256: RDOPN3: LD      A,7     ;Bad Allocation Error
  257:         SCF
  258:         RET
  259:
  260:
  261: _ROPEN: CALL    FCBSCH
  262:         RET     C
  263:         LD      A,8     ;File Not Fonnd
  264:         SCF
  265:         RET     NZ
  266:         PUSH    HL
  267:         LD      DE,(_IBFAD)
  268:         LD      BC,20H
  269:         LDIR
  270:         POP     HL
  271:         LD      A,(HL)
  272:         CALL    FMCHK
  273:         RET
  274:
  275: ;
  276: ; FCB SEARCH
  277: ;
  278: FCBSCH: LD      C,16            ;Directory Lengh
  279:         LD      DE,(_DIRPS)     ;Directory start
  280: FCBSC1: LD      (DEBUF),DE
  281:         LD      HL,(_DTBUF)
  282:         LD      A,1
  283:         CALL    _DRDSB
  284:         RET     C
  285:         LD      B,8
  286: FCBSC2: LD      (HLBUF),HL
  287:         LD      A,(HL)
  288:         CP      0FFH
  289:         JR      Z,FCBSC4
  290:         OR      A
  291:         JR      Z,FCBSC3
  292:         PUSH    DE
  293:         LD      DE,(_IBFAD)
  294:         CALL    FCOMP
  295:         POP     DE
  296:         JR      Z,FCBSC5
  297: FCBSC3: PUSH    DE
  298:         LD      DE,32
  299:         ADD     HL,DE
  300:         POP     DE
  301:         DJNZ    FCBSC2
  302:         INC     DE
  303:         DEC     C
  304:         JR      NZ,FCBSC1
  305:
  306: FCBSC4: DB      3EH     ; Z = 0
  307: FCBSC5: XOR     A       ; Z = 1
  308:         OR      A
  309:         RET
  310:
  311:
  312: ; File Name Compare
  313: FCOMP:  PUSH    BC
  314:         PUSH    HL
  315:         LD      B,16    ;Directory lengh
  316: FCOMP1: INC     DE
  317:         INC     HL
  318:         LD      A,(DE)
  319:         CP      (HL)
  320:         JR      NZ,FCOMP2
  321:         DJNZ    FCOMP1
  322: FCOMP2: POP     HL
  323:         POP     BC
  324:         RET
  325:
  326: ; FILE MODE CHECK
  327: FMCHK:  PUSH    HL
  328:         AND     87H             ;1000&0111B
  329:         LD      HL,@IBUF
  330:         CP      (HL)
  331:         POP     HL
  332:         RET     Z
  333:         LD      A,6             ;Bad File Mode
  334:         SCF
  335:         RET
  336:
  337:
  338: ;=================
  339: ; INPUT FROM FILE
  340: ;=================
  341: INPUT_: LD      HL,RDPNT
  342:         INC     (HL)
  343:         JR      NZ,INP1
  344:
  345:         CALL    FREAD
  346:         RET     C
  347:         LD      A,(FLPNT)
  348:         INC     A
  349:         LD      (FLPNT),A
  350:
  351: INP1:   LD      HL,DMABF
  352:         LD      A,(RDPNT)
  353:         LD      D,0
  354:         LD      E,A
  355:         ADD     HL,DE
  356:         LD      A,(HL)
  357:         RCF
  358:         RET
  359:
  360: RDPNT:  DS      1
  361:
  362:
  363:
  364: FREAD:  LD      A,(FLPNT)
  365:         LD      B,A
  366:         LD      A,(RC)
  367:         CP      B
  368:         JP      C,RDOPN3        ;Bad Allocation File
  369:
```

▶最近TETRISにハマッてしまって，ソ連もなかなかなかにくいことをやってくれるな，と思っています。当分寝る時間が少なくなりそうだ。　渡辺　宗明（20）山形県

```
370:          LD      A,B
371:          CALL    PNTREC
372:          EX      DE,HL           ; DE = RECORD
373:
374:          LD      A,1
375:          LD      HL,DMABF
376:          CALL    _DRDSB
377:          RET
378:
379:
380:
381: ; FILE POINTER (A) => RECORD NO. (HL)
382: PNTREC: PUSH    AF
383:          PUSH    AF
384:          SRL     A
385:          SRL     A
386:          SRL     A
387:          SRL     A               ;A = A / 16
388:          LD      HL,TBLCLST
389:          LD      D,0
390:          LD      E,A
391:          ADD     HL,DE
392:          LD      A,(HL)
393:          CALL    CLREC
394:          POP     AF
395:          AND     0FH
396:          ADD     A,L
397:          LD      L,A
398:          POP     AF
399:          RET
400:
401:
402: ; FAT READ TO BUFFER
403: RDFAT:  PUSH    DE
404:          PUSH    HL
405:          LD      A,1
406:          LD      DE,(_FATPOS)
407:          LD      HL,(_FATBF)
408:          CALL    _DRDSB
409:          POP     HL
410:          POP     DE
411:          RET
412:
413:
414: ; CLUSTER (A) => RECORD (HL)
415: CLREC:  LD      H,0
416:          LD      L,A
417:          ADD     HL,HL
418:          ADD     HL,HL
419:          ADD     HL,HL
420:          ADD     HL,HL
421:          RET
422:
423: ; RECORD (HL) => CLUSTER (A)
424: RECCL:  PUSH    HL
425:          SRL     H
426:          RR      L       ;HL/2
427:          SRL     H
428:          RR      L       ;HL/4
429:          SRL     H
430:          RR      L       ;HL/8
431:          SRL     H
432:          RR      L       ;HL/16
433:          LD      A,L
434:          POP     HL
435:          RET
436:
437:
438:
439: IMF_SIZE  EQU  37H
440: FILE_BF:DS  IMF_SIZE
441: FLSIZE  EQU  FILE_BF+12H  ;DS 2  ;File Size.
442: FLDTADR  EQU  FILE_BF+14H  ;DS 2  ;Start Address.
443: FLEXADR  EQU  FILE_BF+16H  ;DS 2  ;Exec Address.
444: FSTCLST  EQU  FILE_BF+1EH  ;DS 1  ;First Cluster.
445: TBLCLST  EQU  FILE_BF+20H  ;DS 20H ;Cluster table.
446: FLPNT  EQU  FILE_BF+31H  ;DS 1  ;The FILE Pointor.
447: DEBUF  EQU  FILE_BF+32H  ;DS 2  ;Record  No. Which Have The DIR.
448: HLBUF  EQU  FILE_BF+34H  ;DS 2  ;Address Where On IMFORMATION.
449: RC  EQU  FILE_BF+36H  ;DS 1  ;The Number of Records The File  have.
450:
451:   END  start
```

リスト8　共通部分

```
==================  sos.sim  ==================
  1: ;================================
  2: ; include file for S-OS simulate
  3: ;       Programed By T.I
  4: ;               '87 Nov 22th
  5: ;================================
  6:
  7: rcf     macro
  8:         scf
  9:         ccf
 10:         endm
 11:
 12: pushall macro
 13:         push    hl
 14:         push    de
 15:         push    bc
 16:         endm
 17:
 18: popall  macro
 19:         pop     bc
 20:         pop     de
 21:         pop     hl
 22:         endm
 23:
 24: patch   macro   p,q
 25:         ld      de,(0001h)      ;;get 'wboot' entry
 26:         dec     de
 27:         dec     de
 28:         dec     de              ;;de = bios start address
 29:         ld      hl,q
 30:         add     hl,de
 31:         ld      (p+1),hl
 32: patch   macro   r,s
 33:         ld      hl,s
 34:         add     hl,de
 35:         ld      (r+1),hl
 36:         endm
 37:         endm
 38:
 39: bdos    equ     0005h
 40:
 41:
 42: cpminit:ld      c,31
 43:         call    bdos
 44:         ld      a,(hl)
 45:         inc     hl
 46:         ld      h,(hl)
 47:         ld      l,a
 48:         ld      (cntsec),hl     ;sec pae track
 49:
 50:         patch   seldsk  1bh
 51:         patch   settrk  1eh
 52:         patch   setsec  21h
 53:         patch   setdma  24h
 54:         patch   read    27h
 55:         patch   write   2ah
 56:
 57:         ld      c,1
 58:         call    seldsk
 59:         ret     c       ;hl = DPH of B
 60:
 61:         ld      (hl),0
 62:         inc     hl
 63:         ld      (hl),0  ;clear XLT-B
 64:         ret             ;not have scue
 65:
 66:
 67: ; jump table for bios call
 68: ;   patched from 'cpminit'
 69:
 70: ; Select Drive
 71: seldsk: call    0
 72:         ld      a,h
 73:         or      l
 74:         ret     nz
 75:         ld      a,2     ;Device offline
 76:         scf
 77:         ret
 78:
 79: ; Set Track Number
 80: settrk: jp      0
 81: ; Set Sector Number
 82: setsec: jp      0
 83: ; Set DMA address
 84: setdma: jp      0
 85: ; Read Disk
 86: read:   call    0
 87:         and     a       ;cy=0 & cp 0
 88:         ret     z
 89:         ld      a,1     ;Device I/O error
 90:         scf
 91:         ret
 92:
 93: ; Write DISK
 94: write:  call    0
 95:         and     a       ;cy=0 & cp 0
 96:         ret     z
 97:         ld      a,1     ;Device I/O error
 98:         scf
 99:         ret
100:
101: _pause: pushall
102:         ld      c,6     ;direct console I/O
103:         ld      e,0ffh
104:         call    bdos
105:         cp      3
106:         jp      z,0     ;Warm start
107:         popall
108:         ret
109:
110: _print: pushall
111:         push    af
112:         ld      c,2
113:         ld      e,a
114:         call    bdos
115:         pop     af
116:         popall
117:         ret
118:
119: _prthx: push    af
120:         rrca
121:         rrca
122:         rrca
123:         rrca
124:         call    prthx1
125:         pop     af
126: prthx1: and     0fh
127:         add     a,'0'
128:         cp      '9'+1
129:         jr      c,prthx2
130:         add     a,7
131: prthx2: call    _print
132:         ret
133:
134: _prthl: ld      a,h
135:         call    _prthx
136:         ld      a,l
137:         call    _prthx
138:         ret
139:
140: _hlhex: call    ahex
141:         ld      h,a
142:         call    nc,ahex
143:         ld      l,a
144:         ret
145:
146: ahex:   push    bc
147:         ld      a,(de)
148:         inc     de
149:         call    hex
150:         jr      c,ahex1
151:         rrca
152:         rrca
153:         rrca
154:         rrca
155:         ld      c,a
156:         ld      a,(de)
157:         inc     de
158:         call    hex
159:         jr      c,ahex1
160:         or      c
161: ahex1:  pop     bc
162:         ret
163:
164: hex:    sub     '0'
165:         jr      c,hex2
166:         cp      10
167:         jr      c,hex1
168:         cp      17
169:         jr      c,hex2
170:         sub     7
171:         cp      10h
172: hex1:   rcf
173:         ret
174:
175: hex2:   ld      a,14    ;Bad data
176:         scf
177:         ret
178:
179:
180: _ltnl:  push    af
181:         ld      a,0ah
182:         call    _print
183:         ld      a,0dh
184:         call    _print
185:         pop     af
186:         ret
187:
188: _prnts: push    af
189:         ld      a,' '
190:         call    _print
191:         pop     af
192:         ret
193:
194: _msx:   push    de
195:         push    af
196: msx1:   ld      a,(de)
197:         and     a
198:         jr      z,msx2
199:         call    _print
200:         inc     de
201:         jr      msx1
202: msx2:   pop     af
203:         pop     de
204:         ret
205:
206: _flget: pushall
207:         ld      c,1
208:         call    bdos
209:         popall
210:         ret
211:
212: _getl:  pushall
213:         ld      c,10
214:         ld      de,kbf
215:         call    bdos
216:         ld      a,(kbf+1)
217:         ld      hl,kbf+2
218:         ld      d,0
219:         ld      e,a
220:         add     hl,de
221:         ld      (hl),0  ;end code
222:         popall
223:         ret
224:
225:
226: _FILE:  CALL    FNAME
227:         PUSH    DE
228:         LD      HL,NAMEBF
229:         LD      DE,@IBUF
230:         LD      BC,18
231:         LDIR
232:         POP     DE
233:         CALL    SPCUT
234:         OR      A
235:         RET
236:
237: FNAME:  LD      HL,NAMEBF
238:         LD      (HL),A
239:         INC     HL
240:         LD      B,13
241: FILE2:  CALL    FILE3
242:         LD      A,(DE)
243:         JR      NZ,@00008
244:         LD      A,' '
245:         DEC     DE
246: @00008: CP      '.'
247:         JR      NZ,@00009
248:         LD      A,' '
249:         DEC     DE
250: @00009: LD      (HL),A
251:         INC     DE
252:         INC     HL
253:         DJNZ    FILE2
254:
255:         LD      A,(DE)
256:         CP      '.'
257:         JR      NZ,FILE21
258:         INC     DE
259:
260: FILE21: LD      B,3
261:         CALL    FILE3
262:         LD      A,(DE)
263:         JR      NZ,@0000A
264:         LD      A,' '
265:         DEC     DE
266: @0000A: LD      (HL),A
267:         INC     DE
268:         INC     HL
269:         DJNZ    FILE21+2
270:         LD      (HL),' '
271:         RET
272:
273: FILE3:  PUSH    DE
274:         CALL    SPCUT
275:         LD      A,(DE)
276:         POP     DE
277:         CP      ":"
278:         RET     Z
279:         CP      20H
280:         RET     NC
281:         CP      A
282:         RET
283:
284: TOUPPER:CP      'a'
285:         RET     C
286:         CP      'z'+1
287:         RET     NC
288:         SUB     20H
289:         RET
290:
291: @IBUF:  DS      20H
292: NAMEBF: DS      18
293:
294: CUTLP:  INC     DE
295: SPCUT:  LD      A,(DE)
296:         CP      ' '
297:         JR      Z,CUTLP
298:         RET
299:
300:
301: _IBFAD: DW      @IBUF
302:
303: _DTBUF: DW      dmabf
304: _FATBF: DW      fatbf
305: _DIRPS: DW      0010H
306: _FATPOS:DW      000EH
307: _kbfad: dw      kbf+2
308: kbf:    db      20
309:         ds      21
310: cntsec: ds      2       ;sec per track
311:
312: ;
313: ;
314: ;       READ
315: ;
316: ;
317: _drdsb: pushall
318:         push    af
319:         ld      c,01    ;B drive
320:         call    seldsk
321:         jr      c,read2
322:         pop     af
323:         popall
324:
325:         pushall
326: read1:  push    af
327:         push    de
328:         push    hl
329:         call    get
330:         jr      c,read3
331:         pop     hl
332:         ld      de,100h
333:         add     hl,de
334:         pop     de
335:         inc     de
336:         pop     af
337:         dec     a
338:         jr      nz,read1
339:         and     a       ;cy = 0
340:         popall
341:         ret
342:
343: read3:  pop     hl
344:         pop     de
345: read2:  pop     af
346:         popall
347:         ld      a,2     ;Device Offline
348:         scf
349:         ret
350:
351:
352: get:
353:         push    de
354:         ld      (sv_bfad),hl
355:         ld      b,h
356:         ld      c,l
357:         call    setdma
358:         pop     de
359:
360:         ex      de,hl
361:         add     hl,hl
362: ;       ex      de,hl   ;de * 2
363:
364: ;       ex      de,hl
365:         ld      de,(cntsec)
366:         call    div     ; hl = record(hl) / cntsec(de)
```

▶そりゃないよなあ。６月17日にOh!Xを買って，その日のうちに「AJOY.X」を打ち込んだのに。今日届いた「電脳倶楽部」のなかにちゃんとあるんだもんなあ。泣いちゃうよオレ。

味野　真一（23）群馬県

```
367:         push    de      ; de = record(hl) ¥ cntsec(de)
368:         ld      b,h
369:         ld      c,l
370:         call    settrk
371:         pop     de
372:
373:         ld      b,d
374:         ld      c,e
375:         ld      (sv_sec),bc
376:         call    setsec
377:
378:         call    read
379:         ret     c
380:
381:         ld      hl,(sv_bfad)
382:         ld      de,80h
383:         add     hl,de
384:         ld      b,h
385:         ld      c,l
386:         call    setdma
387:
388:         ld      bc,(sv_sec)
389:         inc     bc
390:         call    setsec
391:         call    read
392:         ret
393:
394: ;
395: ;
396: ;       WRITE
397: ;
398: ;
399: _dwtsb: pushall
400:         push    af
401:         ld      c,01    ;B drive
402:         call    seldsk
403:         ld      a,h
404:         or      l
405:         jr      z,write2
406:         pop     af
407:         popall
408:
409:         pushall
410: write1: push    af
411:         push    de
412:         push    hl
413:         call    put
414:         jr      c,write3
415:         pop     hl
416:         ld      de,100h
417:         add     hl,de
418:         pop     de
419:         inc     de
420:         pop     af
421:         dec     a
422:         jr      nz,write1
423:         and     a       ;cy = 0
424:         popall
425:         ret
426:
427: write3: pop     hl
428:         pop     de
429: write2: pop     af
430:         popall
431:         ld      a,2     ;Device Offline
432:         scf
433:         ret
434:
435: put:    push    de
436:         ld      (sv_bfad),hl
437:         ld      b,h
438:         ld      c,l
439:         call    setdma
440:         pop     de
441:
442:         ex      de,hl
443:         add     hl,hl
444: ;       ex      de,hl   ;de * 2
445:
446: ;       ex      de,hl
447:         ld      de,(cntsec)
448:         call    div     ; hl = record(hl) / cntsec(de)
449:         push    de      ; de = record(hl) ¥ cntsec(de)
450:         ld      b,h
451:         ld      c,l
452:         call    settrk
453:         pop     de
454:
455:         ld      b,d
456:         ld      c,e
457:         ld      (sv_sec),bc
458:         call    setsec
459:
460:         call    write
461:         ret     c
462:
463:         ld      hl,(sv_bfad)
464:         ld      de,80h
465:         add     hl,de
466:         ld      b,h
467:         ld      c,l
468:         call    setdma
469:
470:         ld      bc,(sv_sec)
471:         inc     bc
472:         call    setsec
473:         call    write
474:         ret
475: ;
476: ; hl = hl / de
477: ; de = hl ¥ de
478: ;       from M.L. excise (oh! MZ)
479: div:    push    af
480:         push    bc
481:
482:         ld      b,d
483:         ld      c,e     ;ld bc,de
484:         ld      d,h
485:         ld      e,l     ;ld de,hl
486:         ld      a,16
487:         ld      hl,0
488: div1:   sla     e
489:         rl      d
490:         adc     hl,hl
491:         push    hl
492:         or      a
493:         sbc     hl,bc
494:         pop     hl
495:         jr      c,div2
496:         sbc     hl,bc
497:         inc     de
498: div2:   dec     a
499:         jr      nz,div1
500:         ex      de,hl
501:         pop     bc
502:         pop     af
503:         ret
504:
505:
506: sv_bfad:ds      2
507: sv_sec: ds      2
508:
509:
510: error:  ld      hl,mestbl-2
511:         add     a,a
512:         ld      d,0
513:         ld      e,a
514:         add     hl,de
515:         ld      e,(hl)
516:         inc     hl
517:         ld      d,(hl)
518:         call    _msx
519:         ld      a,7     ;bell
520:         call    _print
521:         call    _ltnl
522:         ret
523:
524:
525: mestbl: dw      err1 ,err2 ,err3 ,err4 ,err5
526:         dw      err6 ,err7 ,err8 ,err9 ,err10
527:         dw      err11,err12,err13,err14
528: err1:   db      'Device I/O Error',0
529: err2:   db      'Device offline',0
530: err3:   db      'Bad file Descripter',0
531: err4:   db      'Write Protected',0
532: err5:   db      'Bad Record',0
533: err6:   db      'Bad file mode',0
534: err7:   db      'Bad Allocation Table',0
535: err8:   db      'File not Found',0
536: err9:   db      'Devicr Full',0
537: err10:  db      'File Already Exists',0
538: err11:  db      'Reserved Feature',0
539: err12:  db      'File not Open',0
540: err13:  db      'Syntax '
541: err14:  db      'Error ',0
542:
543:
544: dmabf:  ds      100h
545: fatbf:  ds      100h
```

リスト9 CRCソースリスト

```
================== crc.mac ==================
  1: ;********************************
  2: ; CRC Check Sum Program
  3: ;       Programed by T.Ishigami
  4: ;         '89 Jun.20th
  5: ;
  6: ;********************************
  7:
  8:         .z80
  9:         aseg
 10:         org     100h
 11:
 12: BOOT    EQU     0
 13: BDOS    EQU     5
 14: DMABF   EQU     80H
 15:
 16: start:  LD      SP,4000h
 17:
 18:         LD      C,15            ;Open File
 19:         LD      DE,5ch
 20:         CALL    BDOS
 21:         CP      0FFH
 22:         JR      Z,OPNERR        ;File not Found
 23:
 24:         LD      HL,100H
 25:         LD      (dmpad),HL      ;== Start Address of File ==
 26: ;
 27: LOOP1:  LD      C,20            ;Read Sequential
 28:         LD      DE,5CH
 29:         CALL    BDOS
 30:         AND     A
 31:         JP      NZ,BOOT
 32:         CALL    DUMP
 33:         JR      LOOP1
 34:
 35: OPNERR: LD      DE,ERRMSG
 36:         CALL    _msx
 37:         JP      BOOT
 38:
 39: ERRMSG: DB      'File not found'
 40:         DB      0DH,0AH,0
 41: ;
 42: ;
 43: ; This Routines are quotation
 44: ;       From Oh! MZ Jan '87
 45: ;
 46: ;
 47: DUMP:   LD      DE,msg2
 48:         CALL    _msX
 49:
 50:         LD      HL,DMABF
 51:         LD      C,16
 52: DUMP1:  CALL    _pause
 53:         CALL    LNDMP
 54:         CALL    _ltnl
 55:         LD      DE,8
 56:         ADD     HL,DE           ;HL = HL + 8
 57:         DEC     C
 58:         JR      NZ,DUMP1
 59:
 60:         CALL    COLSUM
 61:         CALL    _ltnl
 62:         RET
 63: ;
 64: ; 1 line Dump
 65: ;
 66: LNDMP:  PUSH    HL
 67:         LD      HL,(dmpad)
 68:         LD      (dmpad),HL
 69:         LD      DE,8
 70:         ADD     HL,DE
 71:         CALL    _prthl
 72:         POP     HL
 73:         PUSH    HL
 74:         LD      B,8
 75:         LD      E,0             ;Clear Sum
 76: LNDMP1: CALL    _prnts
 77:         LD      A,(HL)
 78:         INC     HL
 79:         PUSH    AF
 80:         CALL    _prthx          ;print data
 81:         POP     AF
 82:         ADD     A,E
 83:         LD      E,A             ;add sum
 84:         DJNZ    LNDMP1
 85:
 86:         LD      A,':'
 87:         CALL    _print
 88:         LD      A,E
 89:         CALL    _prthx          ;print h-sum
 90:         POP     HL
 91:         RET
 92: ;
 93: COLSUM: LD      B,33
 94:         LD      A,'-'
 95: CLSM1:  CALL    _print
 96:         DJNZ    CLSM1
 97:         CALL    _ltnl           ;print "--- ... ---",LF
 98:
 99:         LD      DE,msg3
100:         CALL    _msx            ;print "sum: "
101:         LD      HL,DMABF
102:         LD      C,8
103: CLSM3:  PUSH    HL
104:         LD      B,16
105:         XOR     A               ;clear sum
106: CLSM4:  ADD     A,(HL)          ;add sum
107:         LD      DE,8
108:         ADD     HL,DE           ;HL=HL+8
109:         DJNZ    CLSM4
110:
111:         CALL    _prthx
112:         CALL    _prnts          ;print v-sum
113:         POP     HL
114:         INC     HL
115:         DEC     C
116:         JR      NZ,CLSM3
117:
118:         LD      HL,DMABF
119:         LD      B,128-2
120:         CALL    CRC
121:         CALL    _prthl
122:         RET
123: ;
124: ; Thanks to Mr.Iwai
125: ;
126: ;       G(X)=X^16+X^12+X^5+1
127: ;
128: CRC:    LD      D,(HL)
129:         INC     HL
130:         LD      E,(HL)
131:         INC     HL
132:
133: CRC1:   PUSH    DE              ;first  16 bits
134:         LD      E,80H           ;mask pattern
135:         EXX
136:         POP     HL              ;first  16 bits
137:         EXX
138:
139: CRC2:   LD      A,(HL)          ;load 1 byte
140:         AND     E               ; get bit
141:         JR      Z,CRC3          ; cy = 0
142:         SCF                     ; cy = 1
143:
144: CRC3:   EXX
145:         ADC     HL,HL           ;add the bit to HL
146:         JR      NC,CRC4         ;non 16th bit
147:
148:         LD      A,10H
149:         XOR     H
150:         LD      H,A
151:         LD      A,21H           ; 1021h=X^12+X^5+1
152:         XOR     L
153:         LD      L,A             ; HL=HL XOR 1021H
154:
155: CRC4:   EXX
156:         RRC     E               ;rotate mask pattern
157:                                 ;       to get next bit
158:         JR      NC,CRC2         ;case of loop =< 8
159:         INC     HL              ;next bite
160:         DJNZ    CRC2
161:         EXX
162:         RET
163:
164:
165: rcf     macro
166:         scf
167:         ccf
168:         endm
169:
170: pushall macro
171:         push    hl
172:         push    de
173:         push    bc
174:         endm
175:
176: popall  macro
177:         pop     bc
178:         pop     de
179:         pop     hl
180:         endm
181:
182:
183: _pause: pushall
184:         ld      c,6     ;direct console I/O
185:         ld      e,0ffh
186:         call    bdos
187:         cp      3
188:         jp      z,0     ;Warm start
189:         popall
190:         ret
191:
192: _print: pushall
193:         push    af
194:         ld      c,2
195:         ld      e,a
196:         call    bdos
197:         pop     af
198:         popall
199:         ret
200:
201: _prthx: push    af
202:         rrca
203:         rrca
204:         rrca
205:         rrca
206:         call    prthx1
207:         pop     af
208: prthx1: and     0fh
209:         add     a,'0'
210:         cp      '9'+1
211:         jr      c,prthx2
212:         add     a,7
213: prthx2: call    _print
214:         ret
215:
216: _prthl: ld      a,h
217:         call    _prthx
218:         ld      a,l
219:         call    _prthx
220:         ret
221:
222: _ltnl:  push    af
223:         ld      a,0Dh
224:         call    _print
225:         ld      a,0Ah
226:         call    _print
227:         pop     af
228:         ret
229:
230: _prnts: push    af
231:         ld      a,' '
232:         call    _print
233:         pop     af
234:         ret
235:
236: _msx:   push    de
237:         push    af
238: msx1:   ld      a,(de)
239:         and     a
240:         jr      z,msx2
241:         call    _print
242:         inc     de
243:         jr      msx1
244: msx2:   pop     af
245:         pop     de
246:         ret
247:
248: msg1:   db      'INPUT END ADDRESS :',0
249: msg2:   db      0dh,0ah
250:         db      'ADRS +0 +1 +2 +3 +4 +5 +6 +7 Sum'
251:         DB      0dH,0aH,0
252: msg3:   db      'SUM: ',0
253:
254: dmpad:  ds      2
255:
256:         end
```

※今月はスペースの都合によりINDEXの掲載は省略させていただきます。

QUESTION and ANSWER

X68000ユーザーです。X1などのSCRN$を使っているプログラムを移植しようとしても，X-BASICにはSCRN$がないのでうまく移植できません。BGを使えばできるのは知っているのですが，テキスト画面しか使っていないプログラムを移植するのに，いちいちBGを使うのは大変です。X-BASICやXCなどで，テキスト画面から文字を読み込む方法か，外部関数の作り方を教えてください。

静岡県　村田　充昭

HuBASICでのSCRN$は，

A$＝SCRN$(x,y,n)

のように書いて，x，yにテキスト画面上のx座標，y座標，nに文字数を指定して使います。A$にはテキスト画面上の位置（x，y）から半角文字でn個分の文字が代入されることになります。外部関数の作り方は別の機会に譲るとして，今回はBASICでこの問題を解決する方法を考えてみることにしましょう。

ここでは，もっともポピュラーな方法でわかりやすい仮想画面を使ってみることにします。その前に，仮想画面について若干説明をしておきましょう。

知っている方も多いこととは思いますが，ディスプレイに表示されているテキスト画面の内容はテキストVRAMと呼ばれる部分に記憶されています。この部分に文字のASCIIコードを書き込むと，画面上の対応する場所に文字が表示されるという仕組みになっています。テキストVRAMなどの内容の一部，あるいは全部をそのまま（もしくは圧縮などして加工したもの）を別の記憶領域にコピーしたもののことを，仮想画面といいます。当然テキストVRAMと内容は同じでも，あくまでもテキストVRAMのコピーなので，この部分にASCIIコードを書き込んでも画面を書き換えることがありません。それゆえ“仮想”画面なのです。ちなみに，シューティングゲームの衝突判定とか敵の移動なんかが（スプライトやG-RAMの）仮想画面で行われることが多いようです。

さてここでも仮想画面を使って文字の情報を残してやるようにしましょう。96×30のスクリーンモードの場合なら，要素が(96,30）である2次元配列を用意します。画面に文字を表示するときには，その都度X，Y座標に対応する配列にその文字を格納するようにします。

そこで，リスト1のようなサンプルプログラムを作ってみました。20行で宣言しているdsp(95,29)を仮想画面領域としています。このプログラムでは画面サイズを横96桁にするようにしているのでこうなってますが，もし64桁ならdsp(63,29)にしてもかまいません。

関数CLRは仮想画面領域を初期化するもので，画面を初期化するCLS，WIDTH，CONSOLE文などを使ったときは必ずCLR( )を実行しなくてはいけません。100行はaに代入した文字列を画面にプリントする部分です。ここで注意してもらいたいのはprint aとしているのではなくprt(a)となっているところです。

140行から定義されている関数，PRTは引数として渡された文字列を左から1文字ずつ取りだして仮想画面領域に保存し(200行)画面に文字を表示しています(220行)。210行で条件式が成立したときに使っている関数SCRLは画面の最下行に文字を表示して，もし画面がスクロールしたときには仮想画面領域の内容も画面と一緒にスクロールさせるためのものです。

さて，150行に戻りましょう。関数SCRNはHuBASICのSCRN$と同じ書式で使います。このプログラムを実行すると，画面中央に“test～”と表示され，ホームポジション以降には，関数SCRNで得た文字列が表示されます。

このプログラムの実行速度は十分に速いとはいえません。もともとマシン語で書かれるべきシステムルーチンと同じことをBASICで記述しているのですから，これも当然です。ゲームで使うときにはコンパイルが必要でしょう。それにかなり手を抜いて作ってあるので，関数SCRNを使うときなど，X，Y，Nに変な数字を入れて使うと誤動作します。

BASICレベルでゲームなどを移植しようというなら，やはりスプライトのBGを使うことをおすすめします。もともとX68000のテキストVRAMはこのような用途に使用することはあまり考慮されていません。テキストとはいえ，8ビット機でいえば，G-RAMと等価なものですから。ゲームなどには専用に付け加えられたスプライト機能を使うべきでしょう。

それにX1やMZ-2500などでSCRN$が使われる場合はたいていPCGとして使われていますので，カラーの使えるBGのほうが有利だと思います。よく使うものは関数化しておけば，それほど手間が増えるということもないでしょう。画面桁数がどうしても足りないというときは先に挙げた仮想画面を使うようにしてください。

BGでは文字の定義がされていないので，パターンを起こさなければなりませんが，たとえば，アフターバーナーを起動後リセットするなどすれば，英数字に関してはスプライト定義の手間が省けます。

そういうわけですから，皆さんはこれを参考にして自分なりにどんどん使いやすいように改良していくなり，BGのテクニックを磨くなりしていってください。移植する

リスト1

```
 10 int   line_temp,z
 20 char  dsp(95,29)
 30 str   a[255]
 40 /* initialize */
 50 width 96:console 0,30,1
 60 for z=0 to 25
 70 a=a+"+  test -"
 80 next
 90 clr()
100 locate 46,28:prt(a)
110 locate  0, 0:print scrn(0,28,9)
120 locate  0, 1:print scrn(0,29,9)
130 end
140 /* print */
150 func prt(moji;str)
160 int i,x,y,y0
170 for i=1 to strlen(moji)
180   x=pos:y0=csrlin
190   y=(y0+line_temp) mod 30
200   dsp(x,y)=asc(mid$(moji,i,1))
210   if x=95 and y=29 then scrl()
220   print mid$(moji,i,1);
230 next
240 endfunc
250 /* scroll */
260 func scrl()
270 int i,j
280 line_temp=(line_temp+1) mod 30
290 /
300 j=(line_temp+31)mod 30
310 for i=0 to 95
320   dsp(i,j)=' '
330 next
340 endfunc
350 /* clear */
360 func clr()
370 int i,j
380   for i=0 to 95
390     for j=0 to 29
400       dsp(i,j)=' '
410     next
420   next
430 endfunc
440 /* scrn$ */
450 func str scrn(x,y,n)
460 int i:str b
470   y=(y+line_temp) mod 30
480 for i=1 to n
490   if x=95 and y<>29 then {
500   b=b+dsp(x,y):x=0:y=(y+1)mod 30
510 }
520   b=b+chr$(dsp(x,y))
530   x=x+1
540 next
550 return(b)
560 endfunc
```

プログラムがスクロールを考えなくてもいいのなら，その部分はなくてもいいんだし，ほかの部分も必要最低限な機能さえあればことは足ります。そうすることによって考える力がついていくし，自分でプログラムを作るコンピュータならではの醍醐味を味わうことができるのです。

リスト2

```
0000                 1 ;------------------------------------------
0000                 2 ;
0000                 3 ;  ＬＦとＦＦをＸ１につける
0000                 4 ;
0000                 5 ;              Programed by H.Kageyama
0000                 6 ;
0000                 7 ;------------------------------------------
0000                 8
0000                 9 PRINT : EQU     0013H
0000                10 MES:    EQU     000BH
0000                11         ;
F000                12         ORG     0F000H
F000                13
F000                14 ;********************************
F000                15 ;  コントロールテーブルを設定
F000                16 ;********************************
F000                17 INIT
F000 21 0D F0       18         LD      HL,LF
F003 22 89 00       19         LD      (0089H),HL      ; CTRL-P
F006 21 12 F0       20         LD      HL,FF
F009 22 8B 00       21         LD      (008BH),HL      ; CTRL-Q
F00C C9             22         RET
F00D                23 ;********************************
F00D                24 ;         CTRL-P で 改行
F00D                25 ;********************************
F00D                26 LF
F00D 3E 0A          27         LD      A,0AH   ; LF CODE
F00F C3 17 F0       28         JP      LPRNT
F012                29         ;
F012                30 ;********************************
F012                31 ;       CTRL-Q で 改ページ
F012                32 ;********************************
F012                33 FF
F012 3E 0C          34         LD      A,0CH   ; FF CODE
F014 C3 17 F0       35         JP      LPRNT
F017                36         ;
F017                37 ;********************************
F017                38 ;     プリンタにデータを送る
F017                39 ;       A Reg.に出力データ
F017                40 ;********************************
F017                41 LPRNT
F017 F5             42         PUSH    AF
F018                43         ;
F018 01 01 1A       44         LD      BC,1A01H
F01B 21 00 10       45         LD      HL,1000H
F01E                46 LPRNT1
F01E ED 78          47         IN      A,(C)
F020 E6 08          48         AND     08H     ; NOT BUSY
F022 28 10          49         JR      Z,LPRNT2
F024 2B             50         DEC     HL
F025 7C             51         LD      A,H
F026 B5             52         OR      L
F027 20 F5          53         JR      NZ,LPRNT1
F029 CD 42 F0       54         CALL    LTNL
F02C 11 47 F0       55         LD      DE,MESSAGE
F02F CD 0B 00       56         CALL    MES     ; Device offline
F032 F1             57         POP     AF
F033 C9             58         RET
F034                59         ;
F034                60 LPRNT2
F034 F1             61         POP     AF
F035 0D             62         DEC     C       ; BC=1A00H
F036 ED 79          63         OUT     (C),A
F038                64         ;
F038 0E 03          65         LD      C,3     ; BC=1A03H
F03A 3E 0E          66         LD      A,0EH
F03C ED 79          67         OUT     (C),A   ; PC7 = 0
F03E 3C             68         INC     A
F03F ED 79          69         OUT     (C),A   ; PC7 = 1
F041                70         ;
F041 C9             71         RET
F042                72         ;
F042                73 LTNL
F042 3E 0D          74         LD      A,0DH
F044 CD 13 00       75         CALL    PRINT
F047                76 ;
F047 07             77 MESSAGE DB      07H
F048 44 65 76       78         DM      "Device offline"
F04B 69 63 65
F04E 20 6F 66
F051 66 6C 69
F054 6E 65
F056 0D 00          79         DB      0DH:00H
F058 C9             80         RET
```

マシン語プログラムをメモリ上に常駐させて，そのプログラムを必要なときにコントロールキーやファンクションキーで実行することができるでしょうか。もしできるのなら，その方法を教えてください。マシンはX1です。

滋賀県　安田　修

CZ-8FB01 (X1BASIC) のIOCSにはコントロールキーのジャンプテーブルが用意されていますので，そのジャンプアドレスを書き換えることによって簡単に実現できます。0069Hから009EHまで2バイトごとに，CTRL-@，CTRL-A，B，C……Zの順でジャンプアドレスが上位下位が逆になって格納されています。たとえば，モニタに入って0069Hからの2バイトを00H，C0Hに書き換えると，CTRL-@を押すとJUMP C000Hが実行されたことになります。なお，ジャンプテーブルを書き換える際には，BASICで使用されているコントロールキーは書き換えないほうが賢明です。誤動作の可能性があります。

参考までにコントロールキーを使ったプログラムを紹介しましょう。プログラムは0F000Hから置かれるので，

LIMIT &HF000

を実行してプログラム領域を確保してください。

CALL &HF000

で実行です。CTRL-Pでプリンタの改行，CTRL-Qでプリンタの改ページを行うことができるようになります。プログラム自体は非常に短いのでわかりやすいと思います。あまり役に立たないものですが，X68000にも似たようなプリンタコントロールキーがあったので真似して作ってみました。暇な方は入力してみてください。（影山　裕昭）

質問にお答えします

日ごろ疑問に思っていること，どんなことでも結構です。どんどんお便りください。難問，奇問，編集室が総力を上げてお答えいたします。ただし，お寄せいただいているものの中には，マニュアルを読めばすぐに回答が得られるようなものも多々あります。最低限，マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名，システム構成，必要なら図も入れてこと細かに書いてください。また，返信用切手同封の質問をよく受けますが，原則として，質問には本誌上でお答えすることになっていますのでご了承ください。なお，質問の内容について，直接問い合わせることもありますので，電話番号も明記してくださいね。

宛先：〒102 東京都千代田区
九段南2-3-26井関ビル
㈱日本ソフトバンク出版部
「Oh！X質問箱」係

FILES Oh!X

このインデックスは，タイトル，注記——筆者名，誌名，月号，ページで構成されています。梅雨も明けて，さあ夏休み！　ふだんできないこともいろいろ計画できる貴重なときですね。

参考文献
I/O　工学社
ASAHIパソコン　朝日新聞社
ASCII　アスキー
テクノポリス　徳間書店
POPCOM　小学館
マイコン　電波新聞社
マイコン BASIC Magazine　電波新聞社
LOGIN　アスキー

一般

▶ NEW PRODUCTS
　世界初の14インチカラー液晶ディスプレイを搭載したシャープのAXパソコン。——編集部，I/O，7月号，156p.

▶ BIG NEW PRODUCTS
　シャープのアナログジョイスティックの紹介とマウスとして扱うためのドライバ。——市原昌文，I/O，7月号，188-190pp.

▶第68回ビジネスショウ
　ビジネスショウレポート。カラー液晶ディスプレイなど。——編集部，ASCII，7月号，230-231pp.

▶マイクロコンピュータショウ'89
　マイコンショウレポート。——編集部，ASCII，7月号，232p.

▶ CGセミナー
　スキャナを使って原画を取り込む際の注意点などについて。——編集部，テクノポリス，7月号，87-90pp.

▶ビジネスマンの情報管理術
　電子手帳用プログラムBASICカードの紹介と使用例。——塚田洋一，マイコン，7月号，296-299pp.

▶電子手帳インターフェース
　HAL研究所の電子手帳用インタフェイスの紹介。——丹治佐一，マイコン，7月号，300-302pp.

▶なんでもQ&A　シャープMZシリーズ編
　AX286Lが搭載可能なICカードインタフェイスに関する質問。——編集部，マイコン，7月号，372-373pp.

▶記憶媒体の構造を知ろう
　フロッピーディスク，ハードディスクの構造，仕組みについて解説。——丹治左一，マイコンBASIC Magazine，7月号，43-46pp.

▶ HOT INFORMATION
　シャープの時計つき電子メモPA-380と，開発中の32ビットAXワークステーションを紹介。——編集部，マイコンBASIC Magazine，7月号，326p.

▶図解世界のコンピュータちゃん　第14回
　エンジニアリングワークステーションHP9000を紹介。——編集部，LOGIN，11号，174-175pp.

▶図解世界のコンピュータちゃん　第15回
　RISCコンセプトワークステーションAViiONを紹介。——編集部，LOGIN，12号，174-175pp.

MZ-80K/C/1200/700/1500

MZ-700/1500

▶ CHECK！CHECK！CHECK！
　紛らわしい文字を見分けるワンキーゲーム。——まっぴ，マイコンBASIC Magazine，7月号，128-129pp.

▶ FALL BLOCK
　TETRISをヒントにしたブロックゲーム。——藤崎勇一，マイコンBASIC Magazine，7月号，130-131pp.

MZ-1500

▶誌上公開質問状
　MZ-1500のマニュアルを購入するにはどうするか，などの質問に答えている。——編集部，マイコンBASIC Magazine，7月号，65p.

▶移植版，週刊誌を倒せ！
　悪徳週刊誌のビルをヘリを使って鉄球を落として壊す，ワンキーアクションゲーム。——ぱそきちぶーたろう，マイコンBASIC Magazine，7月号，132p.

MZ-80B/2000/2500/2800

MZ-2000/2500

▶移植版Explorer
　あなたは探検家。巨大なピラミッドの中で冒険を繰り広げるアクションパズルゲーム。——山下貴史，マイコンBASIC Magazine，7月号，133-135pp.

MZ-2500

▶今宵の星は
　雀と星の物語ゲーム（？）。——コエダ家具，マイコンBASIC Magazine，7月号，136-138pp.

▶家計簿V1.5
　実用プログラム。——萩原大輔，マイコンBASIC Magazine，7月号，186-187pp.

X1/X1turbo/Z

X1シリーズ

▶ GAMING WORLD
　トイポップ，ガルフストリームを紹介している。——編集部，テクノポリス，7月号，25-45pp.

▶今月の注目ソフト
　リボルティー2についてレビューしている。——相川春利，マイコン，7月号，206-207pp.

▶勉強用マシン語アセンブラ
　X1用のBASICに操作感覚を似せたアセンブラ。@キーによるヘルプ機能など，初心者への考慮がなされている。——赤岩英明，マイコン，7月号，235-240pp.

▶なんでもQ&A　シャープX1/X1turbo/X68000シリーズ編
　アンダーバーの入ったファイル名をもつCP/MファイルをturboCP/Mで読み込むには。——編集部，マイコン，7月号，371p.

▶誌上公開質問状
　X1Fmodel 10にCZ-31Fを接続できるか，X1turboZにヘッドホンをつなぐには，などの質問に答えている。——

　床に就こうと灯かりを消すと，部屋中で赤や緑の目が輝き，彼らに侵されつつある闇に同情する。一瞬たりとも休むことなく輝き続けているLED。本書はそんな誰の部屋でも数十個は活躍しているLED（昔風には発光ダイオード）だけの本。著者はさすがメーカーの技術者だけあって，原理を述べながらもさりげなく，波長（つまりは色ね）を変えたい時のドーピングあれこれとか，何色のLEDにはどういう元素が使われているかなど現場の感覚が顔を出していて，専門用語が頻出しても読ませてくれます。後半は，用途別使用例（回路図付）が中心。最近馴染み深いリモコン用赤外線LEDもリモコンの原理から型番や発光指向特性まで押さえているほか，話は半導体レーザーにまで発展していきます。レーザーにまで回路図をつけてその気のある人になら簡単な発光回路が作れるくらい。全編通した具体的な内容に技術者のこだわりを感じます。

　なお，どこからどう読んでもいいように書かれているので，夜の部屋を支配するLEDの魅力を知るにはよいかもしれません。（K）

やさしいLEDのはなし　伊東新太郎著
日本放送出版協会刊　☎03(464)7311
A5判　220ページ　1,650円

編集部，マイコンBASIC Magazine，7月号，64-65pp.
▶GET AWAY
モンスターに捕まらないように鍵を3つ集めて扉へ向かうパズルゲーム。──クスクス＝チンケ，マイコンBASIC Magazine，7月号，168-169pp.
▶ゲッ！　イングランド
イギリスから脱出するために，すべての敵を倒すか画面から追い出す。──ズオ，マイコンBASIC Magazine，7月号，170-172pp.
▶SAMPLING SYSTEM
X1で音声サンプリングをしてしまうユーティリティ。──月岡拓也，マイコンBASIC Magazine，7月号，188-190pp.
▶サンダークロス─1面─
ゲームミュージックプログラム。──上田順一，マイコンBASIC Magazine，7月号，200-201pp.

X1turboシリーズ

▶月刊ソーサリアンニュース
7月下旬発売予定のソーサリアンシナリオ4を紹介している。──編集部，テクノポリス，7月号，22-23pp.
▶SOFT RADAR
ソーサリアンシナリオ4「宇宙からの訪問者」を紹介している。──編集部，POPCOM，7月号，30-31pp.
▶性格判断
1981年11月号に掲載されたMZ-80K/C用からの移植版。質問に答える自己診断方式で，結果はプリントアウトできる。──折原美昭，マイコン，7月号，241-244pp.
▶移植版とりでの攻防
ストラテジックシミュレーションゲーム。角度を入力することでミサイルを誘導し，相手の砦を早く破壊する。──須賀仁，マイコンBASIC Magazine，7月号，173-174pp.
▶NEW SOFT
開発中のアクションアドベンチャーゲーム，ガルフストリームを紹介している。──編集部，LOGIN，11号，20p.
▶NEW SOFT
今夜も朝までPOWERFULまあじゃんデータ集Vol.2を紹介。──編集部，LOGIN，12号，19p.

X68000

▶X-BASIC拡張プログラム2
文字列処理やマウス，算術機能を拡張する関数集。──WIZARD N，I/O，7月号，191-201pp.
▶サンダーフォースII無敵プログラム
コンフィギュレーションの拡張と，自機数が増減しないモードを作成するユーティリティ。──市原昌文，I/O，7月号，247-249pp.
▶C-TRACE68でSFXふうの絵を描く
レイトレーシングとそのX68000用代表的ソフトC-TRACE68の概要，また作品ができるまでを紹介している。──国友正彦，ASAHIパソコン，16号，94-100pp.
▶AV WORKSHOP
CARD PRO-68K上のシステム手帳用リフィルフォーマット集の発売，OS-9/X68000用AV-RIDERのソースコード掲載などのお知らせ。──編集部，ASCII，7月号，337-344pp.
▶テキスト表示ユーティリティLESS
テキストファイルの内容を表示するためのOS-9/X68000上のユーティリティ。MICROWARE Cが必要。──野出泰史，ASCII，7月号，363p.（リスト425-429pp.）
▶日本語スクリーンエディタED
OS-9/X68000上で動作する漢字対応スクリーンエディタ。X68000のキーボードの特殊キーを生かして作られている。──龍野光照，ASCII，7月号，364-368pp.
▶夢幻戦士ヴァリスII
11月ごろ発売予定の夢幻戦士ヴァリスIIをビジュアルシーンを中心に紹介。──編集部，テクノポリス，7月号，12-13pp.
▶GAMING WORLD
アフターバーナー，J.B.ハロルドシリーズ第3弾DCコネクション，大江戸繁盛記，ソフトでハードな物語2などの新作ゲームを紹介。──編集部，テクノポリス，7月号，25-45pp.
▶SOFT RADAR
発売予定のアルビオンを紹介している。──編集部，POPCOM，7月号，16-17pp.
▶X68000ワールド
ソフトでハードな物語2，R-TYPE，ZAVAS，アフターバーナー，スタークルーザー，ファンタジーゾーン，ジェノサイド，Musicstudio PRO-68K用のデータ集，Human68k Ver2.0の紹介と，マイコンショウのレポート。──編集部，POPCOM，7月号，105-115pp.
▶76th STAR
レベッカの曲のミュージックプログラム。──立間克志，POPCOM，7月号，195-197pp.
▶今月のおススメソフト
X68000版スタークルーザーの紹介。──MUNEPI，マイコン，7月号，203-205pp.
▶X68000マシン語入門
X68000でRS-232Cを使う方法について。今回は，オリジナルの超簡易ターミナルソフトに，アップロード，ダウンロード，行入力機能を追加する。──高橋雄一，マイコン，7月号，336-340pp.
▶なんでもQ＆A　シャープX1/X1turbo/X68000シリーズ編
Human68k Ver2.0のキーボードコントロールおよびバッチエリアサイズについてなど。──編集部，マイコン，7月号，370-371pp.
▶誌上公開質問状
X68000 ACE-HDのハードディスクをいくつかのブロックに分けて起動する場合に，システムをどのように選択するか，などについて。──編集部，マイコンBASIC Magazine，7月号，65p.
▶ジャンピング・バスケットボール
バレーとバスケットがいっしょになったゲーム。2人でプレイする。──土田太郎，マイコンBASIC Magazine，7月号，175-177pp.
▶THEトイレ
画面を動かして赤ちゃんをトイレへ誘導する。──林直貴，マイコンBASIC Magazine，7月号，178-180pp.
▶グラディウスII─A Shooting Star─
ゲームミュージックプログラム。──川野俊充，マイコンBASIC Magazine，7月号，192-194pp.
▶チャレンジ！　X68000
R-TYPE，ニュージーランドストーリー，ジェノサイドなどの新作ゲームを紹介。──佐久間亮介，マイコンBASIC Magazine，7月号，276-277pp.
▶X68000新聞
スタークルーザー，ライトニングバッカス，アークスPRO-68K，大海令，A-JAX，ジェノサイドなどの最新ゲームを紹介している。──編集部，LOGIN，11号，176-181pp.
▶SOFTWARE REVIEW
太平洋の嵐DXを徹底解析。──仮面ライターDX，LOGIN，12号，36-37pp.
▶X68000新聞
Z'sSTAFF PRO-68K Ver2，ねじ式，デジタルクラフト，アフターバーナー，今夜も朝までPOWERFULまあじゃん2データ集，R-TYPE，Temple Masterを紹介。──編集部，LOGIN，12号，176-181pp.

ポケコン

PC-1245
▶THREE DRAGONS
竜を倒すRPG。──田中栄造，マイコンBASIC Magazine，7月号，183p.

PC-1600K
▶ポケコン電子手帳（3）
ポケットコンピュータで電子手帳的な機能を実現する。今回はRAMファイルを使ったファイル機能の作成。──塚田洋一，マイコン，7月号，331-335pp.

PC-E200
▶人ならべゲーム
人を操作して障害物や自分の通った跡を避けながらゴールへ向かうゲーム。PC-14系からの移植版で面数4。──平井真二，I/O，7月号，142-143pp.

機械じかけの脳をつくる
例によってコンピュータと人間の脳という分野の話。人工知能（AI）という言葉ができて30余年がたつ。大流行の一方でシリコンチップの限界に対する危機感からある種の閉塞状況が生まれているAI。それと並行して注目を集めているニューラルネットとバイオコンピュータ。本書ではそうしたAIのたどってきた道筋と，米国での最前線の研究動向とに焦点を当てている。研究者によるAIの入門書といえよう。
スコット・ラッド著　大江秀房訳　啓学出版刊
A5判　348ページ　1800円　☎03(233)3795

ベーシック/コンピュータ入門
目次に続き本書の構成がフローチャートになっていて，どんなふうに話が進むか，ポイントはどこかなどが示されている。2進法の話，記憶装置の構造やマシン語，マンマシンインタフェイスについてなどが簡潔にわかりやすく書かれており，読みものとしても手頃な，初心者向けのタイトルどおりの入門書。プリンタ，ディスプレイ，ソフトウェアなどの言葉がきちんと定義されていて面白い。
西田修著　日本経済新聞社刊
B5判　176ページ　777円　☎03(270)0251

とくに表記したもの以外，価格は消費税

愛読者プレゼント

とじ込みのアンケートはがきの該当項目をすべてご記入のうえ，希望するプレゼント番号をはがき右下のスペースにひとつ記入してお申し込みください。締め切りは1989年８月18日の到着分までとします。当選者の発表は1989年10月号で行います。

T・ZONE ☎03(257)2650

T・ZONEオリジナルバッグ 2名

パソコンショップT・ZONEのオリジナルバッグを２名に。Ｂ４サイズで色は黒。

[1] アイレム販売 ☎06(535)4888

R-TYPE

X68000用5"2HD版
7,800円 5名

究極のスクロールシューティングゲームR-TYPEを５名に。ミサイル，ビームの雨あられを心ゆくまで楽しんでください。

スピタル産業 ☎03(251)2918

ジョイコントターボ3

X68000用ジョイカード
2,200円 5名

連射トリガーとノーマルトリガーが独立したジョイカードの新製品，ジョイコントターボ３を５名に。

データウエスト ☎06(968)1236

第4のユニット3

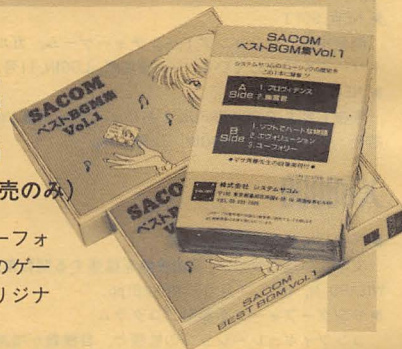
X68000用5"2HD版3枚組
8,800円 3名

可憐なヒロイン・ブロンウィンが活躍する人気シリーズの３作目デュアル・ターゲット。今回はマウスにも対応。

システムサコム ☎03(635)7609

SACOM ベストBGM集 Vol.1 5名

2,000円（消費税込，通信販売のみ）

ソフトでハードな物語，ユーフォリーなど，システムサコムのゲームミュージックを集めたオリジナルテープを５名に。

6月号 プレゼント当選者

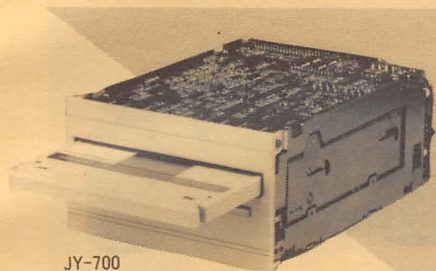
[1] a)ハミングバードオリジナルディスクケース（東京都）山岸祐二　他４名　b)ロードス島戦記ステッカー（大阪府）石田貴志　他９名　[2] マイクロウェアジャパンオリジナルテレホンカード（愛知県）内藤剛克　他４名　[3] 電波新聞社オリジナルポスター（東京都）高橋政秀　他４名　[4] a)光栄オリジナルＣＤ（埼玉県）藤本冬彦　他２名　b)戦国群雄伝ハンドブック（東京都）箭内敬　他２名　[5] a)リバーヒルソフトメモパッド（岡山県）奥山貴士　他４名　b)Ｔシャツ（神奈川県）矢野彰一　[6] スタークルーザーポスター（兵庫県）赤松宏章　他９名　[7] a)NCSオリジナルＴシャツ（宮城県）小野寺光　他１名　b)トレーナー（広島県）平田浩一郎　他１名　c)ブルゾン（愛媛県）　佐伯稔　他１名　[8] 日本ファルコムオリジナルグッズ　a)～g)（兵庫県）宝井和行（千葉県）山中英一（兵庫県）安田功（埼玉県）浅野俊秀　他７名　[9] 九十九電機オリジナル５インチＦＤ（千葉県）西島修　他４名　[10] テクノソフトオリジナルＣＤ（滋賀県）田中真実　他１名　[11] XZRⅡシングルCD（東京都)庄島賢一　[12] ドラゴン（北海道）多田雅一　他２名　[13] Shogun（京都府）大橋広士　他１名　[14] 闘氣王（富山県）狩野太郎　他４名　[15] リボルティーⅡ（北海道）瀬出井賢司　他１名　[16] ザ・キング・オブ・シカゴ（石川県）山下智史　他２名 [17] ソフトでハードな物語２（埼玉県）石田隆志　他２名　[18] 日コン連企画ソフト a)～c)（長野県）五島智明　（愛知県）松野竜明　他４名　[19] ハードゲームソフト　a)～e)（東京都）小西将一　（福岡県）石松一雄　（島根県）山形卓也（群馬県）阿久沢和弘　他14名　[20] たんば（青森県）尾形淳一　他４名　[21] 大海令（秋田県）浅倉和也　他２名　[22] ザ・リターン・オブ・イシター（静岡県）藤田康一　他１名　[23] FINAL Ver.3（茨城県）沢幡智昭　[24] a)サイクロン（愛媛県）石田誠　b)サイクロンアニメキット（奈良県）匂坂真久　他２名　[25] a)WINDEX（岐阜県）今村隆一　b)PP68K（群馬県）中戸太郎

以上の方々が当選されました。おめでとうございます。当選された商品は順次発送致しますが、入荷状況により遅れる場合もあります。また，公正取引委員会の告示により，このプレゼントに当選された方は、この号のほかの懸賞には当選できない場合がありますので，ご了承ください。

PENGUIN INFORMATION CORNER

ペ・ン・ギ・ン・情・報・コ・ー・ナ・ー

NEW PRODUCTS

書き換え可能な光磁気ディスクドライブ
JY-700
シャープ

JY-700

シャープは，ISO規格に準拠した書き換え可能な5インチ光磁気ディスクドライブJY-700を9月から発売する。容量595Mバイト。サンプル価格は500,000円。

新製品は，すでに発売されているJY-500と同様SCSIインタフェイスを採用し，JY-500のアプリケーションソフトを継承しながら，さらに性能の向上を図ったもので，書き換え可能回数1000万回以上，15年間記録・再生ができ，また平均シークタイム60ミリ秒（10mm移動時），データ転送速度925Kバイト/秒という高速データアクセスを実現している。

カートリッジ型の光磁気ディスクはサンプル価格38,000円。オートローディング機構によりディスクドライブへの着脱も容易にできる。ディスクドライブの大きさは幅146×奥行225×高さ82.5mm，重さ約3kg。

<問い合わせ先>
シャープ㈱　☎06(621)1221，03(260)1161

ラップトップワープロ2機種
WD-A300/A330
シャープ

シャープは，ディスプレイにバックライトつき640×400ドットの大型液晶を採用した書院シリーズのラップトップ型パーソナルワープロ2機種を発売した。それぞれ，白黒液晶画面で表計算ソフト書院カルクが標準装備のWD-A330(185,000円)と，青色の液晶画面で書院カルクは別売のWD-A300(165,000円)。

両機種とも，24ドットと16ドットの2種類の表示文字を切り替えられる。プリンタは熱転写/感熱方式で，50文字/秒の高速印字を実現。52ドットと48ドットの高品位文字を使い分けられる。

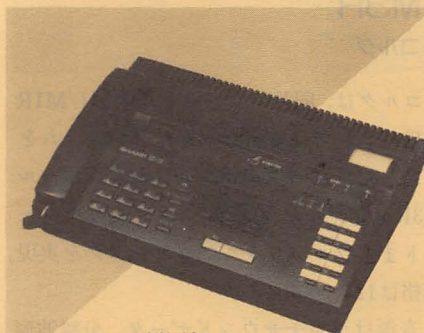
13万語の辞書と4.3万例のAI辞書を持ち，郵便番号辞書も標準装備。

内部メモリはA4原稿で約20枚分，3.5インチFDD1基搭載。オプションでスキャナやカットシート/ハガキフィーダ，各種辞書・用語集，電子手帳通信ケーブル（これで電子手帳のデータを受信できる）などがあり，またカードモデムやMS-DOSテキストファイルコンバータ，RS-232Cインタフェイスなども今後発売される予定。

<問い合わせ先>
シャープ㈱　☎06(621)1221，03(260)1161

WD-A330
B4パーソナルFAX
UX-10
シャープ

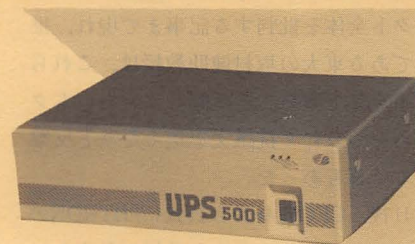
シャープは，B4サイズの読み取り/記録のできるパーソナルファクシミリUX-10を発売した。価格は128,000円。

UX-10は，ビジネス機並の機能を持たせた新製品で，B4読み取り/記録のほか，伝送時間はA4標準原稿で15秒，16階調中間伝送機能，スーパーファインモード機能を持つG3機。

UX-10

電話機能も，電話/ファクシミリ自動切り替え，留守番電話接続のほか，各10カ所のワンタッチダイヤルと短縮ダイヤル，親子電話，転送，内線通話などの各機能を備えている。

大きさは幅368×高さ116×奥行258mm，重さ4.5kg。

<問い合わせ先>
シャープ㈱　☎06(621)1221，03(260)1161

低価格の小型無停電電源装置
UPS-500
スワロー電機

UPS-500

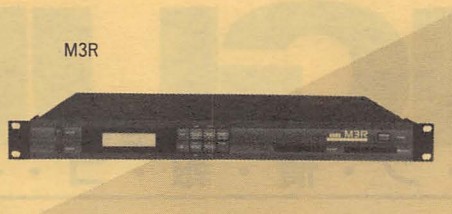
A4サイズの小型無停電電源装置UPS-500が，スワロー電機から発売された。価格は39,800円。

このUPS-500は，1台でパソコン2セットを使用できる。バックアップ時間は8分間（400VA）。停電時（瞬断時も含む）など

の変動から10ミリ秒以下で対応する。ブザー音と発光ダイオードの点滅で停電を知らせ，また外部機器にも停電信号を送る。
<問い合わせ先>
スワロー電機㈱　☎06(714)0431

ハイコストパフォーマンス音源モジュール
M3R
コルグ

コルグは，同社のシンセサイザM1/M1Rに搭載されているaiシンセシスシステムを採用した1UサイズのマルチM音源モジュールM3Rを発売した。音源からステレオエフェクトまでフルデジタルプロセシングを実現。価格は124,000円。

音源はマルチサウンドデータ，分離波形，ドラム波形，DWGSの4タイプ135種を搭載。VDF(バリアブル・デジタルフィルタ)，VDA（バリアブル・デジタルアンプリファイア)，2系統のMDE（ステレオマルチデジタル・エフェクタ）により広範囲な音作りが可能。MDEはそれぞれリバーブ，ディレイ，エキサイター，コーラスなど33種類のエフェクトを内蔵している。レイヤー，スプリット，マルチなどが可能なコンビネーション機能も搭載した。

本体には100種のプログラムと100種のコンビネーションを記憶でき，PCMカード（M1/M1Rと共通，13,000円)，プログラム/コンビネーションカード(8,500円)，RAMカード(11,000円)による各データの追加ができる。

また，オプションのリモートエディタRE1を接続すれば，M3Rの全パラメータを6セクションに分け，8本のスライダーによってアナログ的にコントロールできる。

大きさは幅482×高さ44×奥行332.5mm，重さ4.9kg。
〈問い合わせ先〉
㈱コルグ　☎03(325)5691

デジタルパワーサプライ
DPS-1
日本マランツ

AV機器への電源の安定供給と，複雑化する接続の交通整理，操作の簡略化などを目的としたデジタルパワーサプライDPS-1が，日本マランツから発売された。価格は68,000円。

DPS-1は，パワーアンプ用電源とローシグナル電源とを分離しており，電源回路からのノイズのまわり込みをシャットアウト，クリーンアース化を実現した。パワーアン

DPS-1

Again Watch

BTRON採用断念騒ぎ

6月中旬の日経新聞に，「教育用標準パソコンの仕様策定を進めているコンピュータ教育開発センター(CEC)がBTRONの採用を断念した」という記事が載った。これを機に派生情報が飛びかい，業界から官庁筋までが大騒ぎになった。中にはTRONプロジェクト全体を批判する記事まで現れ，推進役である東大の坂村健助教授は，これらの記事を見て激怒。各新聞社にいちいちクレームをつけたり抗議文を送ったりと反撃をする一幕もあった。

数日後，CEC自身から公式に「BTRONの断念はしていない」という見解が発表され，ようやく収まったが，とにかく最近では珍しい騒ぎだった。

すべての情報をまとめてみると，やはり火のないところに煙はたたないということわざどおり，記事が突然出ただけの動きはCEC側にあったようだ。CECは，これまでハード/ソフト両面にわたる小中学校向け教育用標準パソコンの仕様策定作業を続けてきたが，この仕様からOS，とくにアプリケーションインタフェイスの標準化をやめる決断をこの騒ぎの少し前に行ったらしい。「断念報道は事実無根」というのも間違いなのである。

OSの標準化をやめた理由は，どうやらすでにパソコンを教育用として使い始めている学校の先生から，これまでのソフトが使えないようなOSでは困る，というクレームがついたかららしい。当然といえば当然のクレームである。だからこそCECでは，BTRONだけでなくMS-DOSも併存させる日本電気案を残していたわけだが，それでもまだ不安とあってか，とうとうOS自体の標準化を見送ることにした。

ちなみにこうしたTRONのイメージダウンになる兆しを阻止しようと，推進に最も熱心な松下電器産業は，この翌週に，年内にもBTRONを使った卓上型ワープロを製品化する意向である，との情報をマスコミに流した。やはりBTRONをめぐる動きは面白い。

新次元に突入したラップトップパソコン

6月に2台のエポックメイキングなラップトップパソコンが発表された。

ひとつはセイコーエプソンのPC-286NOTEエグゼクティブ，もうひとつは東芝のダイナブックJ-3100SS。いずれも2kg台と軽く，望まれていたラップトップパソコンを，両社なりに現状でできる限り具現化した製品だ。

いずれもA4判の大きさで，裏ブタが液晶ディスプレイになっている。CPUは8086系。先に発表したエプソンのほうは一連のPC-9801互換機。厚さは35mmで重さは2.2kgと，世界で最も薄くて軽いという。フロッピーはなく，代わりにRAMディスク512Kバイト（最大1.1Mバイト）と，ICカードスロットを用意した。ROM版ソフトやモデムなども内蔵し，価格は45万8000円。

一方，J-3100SSは重さ2.7kg，厚さ37mmとエプソンのものとそう変わらないが，なんといってもこれで3.5インチFDDを1基装備しているのがすごい。メモリも1.5Mバイトと十分で，価格は19万8000円。

どちらも思い切った製品だが，やはり東芝のほうがインパクトが大きい。フロッピーつきといい，価格といい，壮観だ。名前を

プ用には独立したACアウトレットを4つ装備，モノラルパワーアンプ4台を使用したBTL接続にも対応。またシステム全体のマスタースイッチなどで複雑化する接続を整理し，操作を簡略化させている。最大供給可能電力1300W。

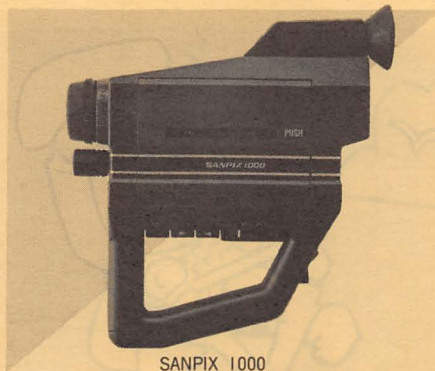
大きさは幅250×高さ84×奥行215mm。重さは7kg。

<問い合わせ先>

日本マランツ(株)　☎03(719)2231

録画・再生デッキ一体型CCDカメラ
SANPIX 1000
興亜物産

興亜物産は，オーディオカセットテープで録画・録音・再生ができる，デッキ一体型のモノクロCCDカメラ，チョイ録ビデオカメラSANPIX 1000 を発売した。価格は48,000円。

新製品は，オートアイリス機能を搭載したCCD素子カメラで，単3電池6本で約2時間作動する。90分用カセットテープで11分(片面各5分30秒)の録画・録音ができ，ま

SANPIX 1000

たデッキ一体型なので付属のケーブルをテレビに接続すれば再生もできる。

大きさは幅220×高さ60×奥行220mm，重さは電池・テープ含め850g。

<問い合わせ先>

興亜物産(株)　☎0535(42)2321

携帯用OHP
PRESES・VP50
リコー

B5判より小さくアタッシュケースに入る携帯用OHP，PRESES・VP50がリコーから発売された。価格は78,000円。

VP50は，透過式の採用で明るい投影画像が得られる，小型で軽量，などの特徴を備えた新製品。使用できるOHPフィルムは，同社のPRESES・VP365で作成したものや，ビデオプリンタから出力したハンディタイプのOHPシートなど。光源はダイクロイックミラーつきハロゲンランプ。投影倍率は7.5～11.8倍，投影距離は0.85～1.3m，投影角度は0～17度。

大きさは収納時で幅170×高さ60×奥行230mm。重さ1.4kg。

<問い合わせ先>

(株)リコー　☎03(479)3111

PRESES・VP50

ラップトップ新時代　1989-08

XEROXが昔思い描いた「ダイナブック」からそのままとっただけのことはある。いよいよラップトップがあるべき姿に到達しつつあるという感が強い。

バッテリー稼働時間が2時間半（エプソンのほうは3時間）だから実際にラップトップで必要な条件をすべて満たしているかといえばそうではないのだが，この際，それは置きたい。

10kgで百万円もするような重くて高いのがあたりまえという，ラップトップの間違った風潮を打ち破ろうとしている価値だけでも大きいのだ。また，8086CPUということで不満のある向きもあるだろう。しかし，今回これだけの製品ができたことで，次回以降をさらに期待すればいいのである。CPUと周辺ICを取り替える作業くらいなんとでもなる。卓上型パソコンを盲目的に持ち運び式にして，誇大広告で売りつけようとするメーカー群に比べれば，東芝の姿勢は本当に立派である。

こうなると，日本電気にもぜひ出てきてほしい。いまや家電業界における松下同様に，パソコン業界における日本電気は，開発は2番手でも一番たくさん売る，という方式を確立してしまっているから台風の目だ。とくに米国では，昨年「ウルトラライト」という超軽量ラップトップ機を先行発売した実績があるから，絶対に近々のうちに対抗馬をぶつけてくるはず。一説によれば，PC-98LTを9801互換にした決定版を開発中ともいわれている。これで日本電気が出揃えば，ラップトップパソコン市場は低価格ワープロなみに爆発的な盛況を見せるようになるかもしれない。久々に，パソコン市場に明るい話題が出てきた。

SHORT AGAIN

キヤノン，NeXTと提携

元アップル社のスティーブン・ジョブス氏率いるNeXT社が，キヤノンと資本，製品開発，販売にわたる広範囲な提携を結んだ。キヤノンは，秋にも日本でNeXTを発売する意向。MacintoshでもMSXでもAXでもうまくいかないキヤノン，NeXTは最後の勝負となるか？

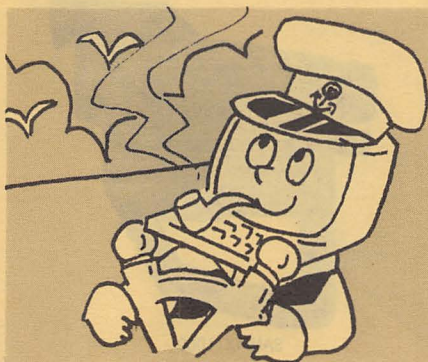
スーパーファミコン，発売延期へ

任天堂がこの秋に販売を予定していた16ビット家庭用テレビゲーム機「スーパーファミコン」の発売が遅れる模様。任天堂では，ゲームボーイと輸出用ファミコンのヒットでマスクROMが不足してソフト供給のメドが立たないと説明しているが，開発が遅れているのでは，との見方もある。さてさて。

任天堂，TETRISで勝訴

この任天堂，米国で争っていたTETRISのサブライセンス権をめぐる裁判で，テンゲン社に勝訴した。

米7社，DRAMメーカーを設立

IBM，DEC，HPのコンピュータメーカー3社と，インテル，ナショナルセミコンダクタ，AMD，LSIロジックの半導体メーカー4社の合計7社は，このほど共同出資して，米国でダイナミックRAMの製造会社を設立することになった。中心となるのはIBM。いよいよIBMが半導体分野の制圧にも乗り出したといえる。2年後をメドに，4MビットDRAMの量産から開始したい意向だ。DRAMは日本のメーカーが圧倒的に優位で8割以上のシェアを持つ。苦境の米国勢が大同団結して対抗しようというプランだが，さてうまくいくかどうか？　(K.T.)

FROM READERS TO THE EDITOR

すっかり梅雨も明け（てればいいんだけど），いよいよ本格的な夏の到来です。でもパソコンユーザーの皆さんは部屋にこもって冷房病にならないように。それから，TOWNSの話題があいかわらず多いけど，そろそろ悪口はやめようよね。

◆今年もあぶない福袋をやるかと思いきや，やりませんでしたねぇ。何か心境の変化でもあったんですか？　四條 智也（19）静岡県

いや〜，年を取るとあぶないことはできなくなっちゃって……うそうそ。

◆“これからのXfamily”を読んで，X68000が発表されたころの大きな衝撃を思い出します。それまでX1の16ビットが出たらどんなものになるのだろうか，ということを話し合っていましたが，発表されると私たちが持っていた夢をはるかに超えたものでした。FM TOWNSは“パソコンが変わる”という宣伝をしてきましたが，X68000もパソコンを変えたマシンであると思います。　谷川 浩司（20）北海道

◆特集に合わせて，これからのX68000に望むことを少し。やはり，仕事に使えるソフトウェアの充実でしょうか。今のままでは「仕事は98，遊びは68」という棲み分けが起きてしまうのではないかと思うのです。特にワープロについては早くなんとかならないでしょうか。ビジネスショウで見たWORD PRO-68Kのデモを見た限りでは，X68000におけるワープロ環境の向上はまだまだ先という気がしたのですが（以前から不思議に思っていたのですが，ツァイトはなぜZ's WORD JG PRO-68Kを出さないのでしょうか）。　荒川 猛（24）東京都

◆“これからのXfamily”非常に面白く拝見しました。読みながら，ウンウンとうなずくばかり。特に“ビジュアルインタフェイスの心”に出てきたチビシェルは私も欲しいと思う。それと電子システム手帳とX68000の接続は早急にお願いしたいものだ。　鯛 富之（26）神奈川県

◆特集の，TORII准将，ISHIMOCHI大佐および金井氏との対談はなかなかよかった。というのも，皆さん自分たちのマシンに誇りを持っていらっしゃるようだから。この人たちならたとえペケロクがドツボに落ちても最後まで付き合ってくれると感じるのです。この感じはとても大事で，富士通はこの点でまったく弱いのです。THE COMPUTER 4月号の記事を読みましたが，富士通のお偉いさんは理想ばかり掲げてちっとも足元を見ていない。この調子だと，TOWNSの雲行きがあやしくなった時点で「あれはステップにすぎません」なんて言い出してもおかしくないと思うのです。　山崎 徹（22）神奈川県

◆Xfamilyのこれからが楽しみである。もっともっと夢が大きくなって現実のものになるのを期待している。　色部 弘之（51）愛知県

ユーザーからこんなに期待されているパソコンてないですよ。シャープさん読んでいますか？　読んでいたら全社をあげてXグループを応援してくださいね。

◆編集室の皆さん，まだ，X1turboファンが何億人といることをお忘れじゃないですか。わかっていたらX1turboの記事をもっともっと増やしてください。お願いしますだー。　島村 哲郎（14）岐阜県

X1/turboはプログラムを送ってくれる人が少ないんですよね。今回のX1特集をきっかけに奮起してくれるといいのですが。

◆粟野さんのハード工作記事は面白く読ませていただきました。これほど細かな点まで手取り足取り書いてある製作記事というのも珍しいですね。少々ハードをかじったことのある人なら，この記事を丁寧に読めば，きっと自分で回路図を書ける立派なハードの鬼になれるでしょう。　属 真人（25）京都府

◆今月の特集の学習リモコンの製作というのを見てすぐに作ろうと思いました。それは現在リモコンが机の上に7個もあるからです。そして，ハードの製作から取り掛かったのですが，赤外線発光ダイオードがなかなか手に入らなかったのです。それと，ジョイスティックのコネクタですが，壊れたジョイスティックのケーブルを取ろうとして思ったのですが，あいにくピンの9番（GND）には線はきていませんでした。だから，私も9ピンのDSUBコネクタを使用しました。ハードのほうはかっこよくできたので今度写真を送ります。　前田 裕史（20）山口県

粟野さんのリモコンは編集部内でも大好評。ついでにバーコードの情報も取り込めたらなんて思うのですが。

◆私はX1twinでPCG定義の高速化に挑戦している。が，45倍速はできたが，93倍速ができない。その筋の人からは，93ってどこから出てくんだぁ，といわれそうですが，93＝（15×2＋1）×3。つまり，8ラインで2つのPCGを（最後を除いて）定義しているのですが，できん……。　坪井 浩（16）神奈川県

◆緊急情報を読みました。ウイルス作成の動機は不正使用ユーザーへの警告とのこと。やれやれ「報復」とは何をか言わんや。両者ともにコンピュータから離れるべきですな。困ったもんだ。　妹尾 義正（38）北海道

◆ウイルスの記事を読みました。どうしてこんないらぬことをするのでしょうか。同じコンピュータ産業に従事するものとして恥ずかしいし，同じX68000ユーザーとして腹が立つ。もっと法律を厳しくして罰するべきだと思います。　中村 佳嗣（22）神奈川県

ウイルスについてはたくさんの意見をいただきました。確かに被害届けを出せば事件として問題になるでしょうが，今はユーザー1人ひとりが認識を深めることが重要でしょう。悪意がなくても子供がナイフを振り回せば立派な凶器になるのですから。

◆6月号166ページの佐藤さん，ダンプリストを究めるのにはブラインドタイプが一番です。私など，ダンプを打ちながら会話をしたり，考え

◀賀内 亨（19）岡山県
大胆な構図でそびえ立つ迫力のX68000に可愛く元気な妖精さん。中古で買っても夢は夢。結構高いけど，やっぱり欲しいよね。

◀大谷 郁夫（17）神奈川県
「あっ，流れ星だっ。よし，願い事を3回言おう。どうか載りますように×3」なんてメッセージが添えてありました。よかったね。

事をすることもあるというレベルに達しました。それでは，ブラインドタイプをマスターする方法ですが，1)ホームポジションを守る。2)キーボードを見ない。3)1日1時間は練習する。これらを守って地道に努力すれば遠からず道は開けるでしょう。信じる者は騙される。いや救われる。何事も努力が肝心です（私は1年以上かかった)。P.S.このごろ“オタク”と“その筋”の比率が逆転してませんか？江原　忠士(19)岡山県

◆台湾製ATARI仕様のジョイカード（for MSX, X68000）に配線ミスがありました。コネクタの8番と9番が逆になっていました。信じ難いことですが，格安品には注意しましょう，ということのようです。　桜井　貴（24）東京都

◆電波新聞社の「XE-1ST」を使っているX1turboユーザーに朗報。ソーサリアンでMODEをSEGAにするとTRIGGER[A]（普段は剣のスイッチ）が「隊列の変更」のスイッチとなるんです。いちいちキーボードのCを押さなくてもよくて便利ですぜ。ほかにも，メガドライブのスペハリIIもSEGAモードで遊べたりする。初期設定を済ませて，ハリアーの後ろ姿が画面に出てから，パッドと差し換える。[A]がポーズ，[B]でショット。おかげでクリアできました。

衣川　慎一（22）愛知県

ありがとう。これからも皆さんからの情報をお待ちしています。

◆5月号16ページの一番大きい写真で，コンパニオンのおねーさんの隣の白い服着たおにーさん。ダメじゃないですか，NECの手先(PC-88ユーザー）がそんなとこに写ってちゃあ。いや，すいません。彼は私の友人でして，その日はC&Cフェアの帰りに，私が強引にパソコンフォーラムに連れてきたのでした。しかし，X1ユーザーの私を差し置いてOh!Xに写真が載ってしまうとは……。　渡辺　光輝（18）埼玉県

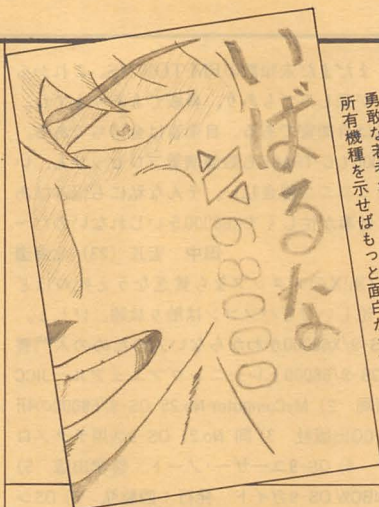
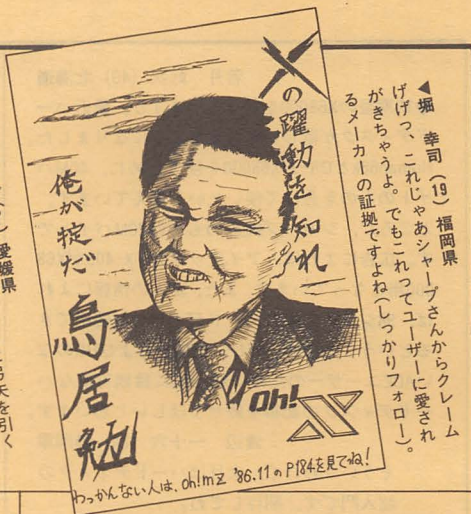
◆小フーガト短調ですか。懐かしいなぁ。この曲，中学の音楽の授業で聴くんですよね。僕はそのとき，とっても感動してX1でプログラムしたんです。6年前になるかなぁ。X1はPSG3声，小フーガは4声だから，足りないところはいちいち響きのいい3声にアレンジして。それから，ブラスバンドやっている友達を呼んできてデバッグしてもらったっけ。感動したなぁ，熱中したなぁ，夢があったなぁ。そして，今日あのときの感動が蘇りました。ありがとうございます。私，まもなく20歳。まだまだ夢もあれば，道もある。　長田　憲司（19）東京都

小フーガの音の深みには感動したでしょう。立川さんのミュージックプログラムにはこれからも期待してくださいね。

◆僕はこの年で「ライブマン」と「マスクマン」の歌が好きでX1turboZで作っています。こんな曲でも受け付けてもらえますか？

林　智広（15）東京都

むしろ，そういうユニークなもののほうがグッドですよ。

◆村田氏の「12語の68000演習プログラム」は今までのマシン語入門記事のなかでは“あっさりしている”という点が功を奏してか，群を抜いてわかりやすい。小学校6年生から7年間，コンピュータをいじり続けているにもかかわらず，マシン語に何度も挫折したトロい僕ですが，なんとかプログラムが組めるようになった。これからもわかりやすい記事を“あっさりと”書き続けてほしい。　中野　哲靖（17）鹿児島県

きっと村田さんの連載はX68000マシン語入門のスタンダードとなると思いますよ。

◆とうとう僕にも「受験生」の肩書きが付くようになりました。長いプログラムはもう打ち込むまいと思っていたのに，悪魔の囁きが……。その名は「組曲ユーフォリー」。僕はユーフォリーの大ファンなので，慌ててMusicBASICから打ち込んでいます。西川善司さんのバカー！（でもうれしい）ちなみに，システムサコムのBGM集も通販で申し込みました。もちろんユーフォリーも入っています。まだ，届いていないんですが楽しみです。わくわく。

角谷　真（17）三重県

◆ときどき買っているもんで，MusicBASICを知りません。ぜひもう一度リストを公開してください。　四方田　賢司（14）埼玉県

MusicBASICは1988年12月号で発表されました。また，今年の5月号でデバッグを兼ねた拡張がされています。まだバックナンバーがあるので利用してくださいね。

◆“娘のために”を口実にYAMAHAのEOSのDS55を買いました。しかし，誰も演奏しないのももったいないからと言いつつ，家内に金を出させてMIDIボードの部品を買い，2～3曲のアニメの主題歌をMMLにて演奏させたところ，息子，娘がえらくお父さんを尊敬してくれました。たまには子供の相手もしないといけないですね。

富崎　雅義（32）徳島県

MIDIボードさえ作れば，X1/turboでも音楽に関しては16ビットマシンに劣ることはありません。お父さん頑張って。

◆息子の章吾（6歳）は私よりゲームが上手です。スペースハリアーは最終面までかるく行きますし，サンダーフォースIIもなかなかのものです。今度はアフターバーナーとサイバースティックを狙っています。最近はOh!Xを見て妹にゲームの説明をしています。私のボーナス時の小遣いは，サイバースティックとHuman68k Ver.2.0に消えてしまいそうです。

白木　準三（37）埼玉県

すごいですね。息子さんは将来，立派なゲーマーになることでしょう。

◆これよ，これ。こういうのを待っていたんだよ。ラベルに数字しか使えないのと，IF文がイマイチなのを除けば，もうこれ以上のものはない。うーん，ナイス。　増田　純一（17）埼玉県

実用本位の巨大な言語より，TTCのほうがかえってプログラミングの楽しさを味わえるかもしれませんね。

◆マンハッタンシェイプは地震のとき大丈夫なんでしょうか？　仕事をしていてときどき心配になるんですが。　佐藤　泰満（30）宮城県

地盤の調査が先決かと思われますが……。

◆私は，すでに100万円以上も貯金があります。親は，「うちの子，ケチでお金を使わないんですよ」と皆に言っていますが，「X68000を買いたいよ」と言ったら猛反対したのはどこの誰なんだよ！　山口　寛憲（19）愛知県

うーん，オバタリアンのネタとして投稿するといいかも。わーっ，ごめんなさい。

◆1988年11月号バックナンバーを入手し，59ページの「S-HCOPY for X1」を読みました。このハードコピールーチンはX1turboでは動作しないのでしょうか。　村松　良彦（54）北海道

縦200ラインのグラフィック画面なら大丈夫ですよ。X1用BASICを使ってくださいね。

◆つ，ついにボクのX1がいかれてしまった。電源を入れて立ち上げると，な，なんとテキスト画面が映らない！　グラフィックは映るのに。おかげで，S-OSをいじることができなくなってしまった。どうにかしなくては。

原　勝則（19）岡山県

マニアタイプは7年前に製造された機種ですからね。でもまだ修理が効くんじゃないかなぁ。お大事にね。

◆ようやくX68000 PRO-HDを，ガンバッテおじさんも買ってしまった。ところがマニュアルで「ハードディスクの使用について」が，初心者なもんで全然わからず，この1週間眠れません。編集部の皆さん，やさしくやさしくおしえて！

◀織田　孝之（?）愛媛県
着々と勢力を伸ばすX68000に弓矢を引く勇敢な若者か。それともただのヒネクレ者か。所有機種を示せばもっと面白かったのに。

◀堀　幸司（19）福岡県
げげっ，これじゃあシャープさんからクレームがきちゃうよ。でもこれってユーザーに愛されるメーカーの証拠ですよね（しっかりフォロー）。

若月 好次(40)北海道

◆初期型のX68000を持っていますが，最近ハードディスクが欲しいと考えるようになりました。Human68kとOS-9/X68000を使うために，40Mバイトのものを分けて使いたいと考えています。ところが，シャープ純正のものは20Mバイトです。広告によれば，アイテックのITX-403がX68000用となっています。また，友人の情報によれば，98シリーズ用のハードディスクが使用できるということです。ぜひ，初期型およびHDのないACEユーザーのためにX68000に接続可能なハードディスクの記事を載せてほしいと思います。

渡辺 一十六(24)静岡県

そういうわけで，来月はハードディスクの超入門です。期待してね。

◆TOWNSの出現はX68000にあまりよく思われていないようだが，大いに喜ばなくてはいけないと思う。X68000もTOWNSも，どっかの会社みたいな考えでなく新しい方向を見出そうとして出現したのだから，そういう点でX68000とTOWNSは共通点を持っているのではないか。この2台がこれからのパソコン界の首領となることを期待して……。

本告 圭(16)熊本県

◆FM TOWNSですが，売れてほしいですネ。これとX68000とで，某DOSマシンでは難しい，新しいファンタスティックな環境が創造していければパソコンの底辺も広がり，パソコンの普及そのものにはいいことだと思うのですが。

郡 茂樹(32)兵庫県

◆98は初めから独走だった。16βもMZ-5500/6500もこけた。人々は彼が1位を走っていたからいいランナーだと決めつけた。今も順調に走っているようだ。その何年かあとにX68000がスタート。88VAも走り始めた。あっ，どうしたことか，88VAリタイヤ，無理があったようです。2年後，FM TOWNSがスタート。速い，パワーがある。人々にやたらと愛敬を振りまいています。X68000を抜かすでしょうか？ X68000選手，98に大差はありますが走りはいたって快調，心強い走りです。98選手は32ビットコーナーを回ってすでに386にタッチ。未だに控え室に姿を見せないX68030選手，初めから飛ばしているFM TOWNS，さてこのレースはどうなるのでしょう。

上野 壮也(18)大阪府

まだまだ未知数のFM TOWNS。これからが楽しみでもあり，脅威でもありますね。

◆私は自衛官である。自衛官は金持ちである。X68000もC-TRACEも数値演算プロセッサも，いつもニコニコ現金払い。そんな私にも悩みはある。仕事が忙しくてX68000をいじれないのだ〜。

田中 宏正(23)北海道

Oh!Xのスタッフなら貧乏なうえ死ぬほど忙しいけどパソコンは触り放題。げしょ。

◆OS-9/X68000がわからない人のための入門書
1) OS-9/68000トレーニングマニュアル JICC出版局 2) MyComputer No.25 OS-9/68000の研究 CQ出版社 3) 同 No.21 OS-9活用テクノロジー 4) OS-9ユーザー・ノート 啓学出版 5) RAINBOW OS-9ガイド 発行：西脇弘 6) OSシリーズ7 OS-9/68000 共立出版
1)と2)があれば初心者でもOS-9のオペレーションができます。

佐藤 栄治(?)東京都

◆私はもともとFM-8ユーザーで，OS-9をずっと使っていました。それが，シャープがX68000を出してくれたのできっとOS-9が出るだろうと思ってX68000を購入しました。ただOS-9がなかなか発売とならなかったので，しかたなくHuman68kでC compiler PRO-68Kを使っていましたが，今度はHuman68kのヒストリ機能の便利さに惚れ込んでしまい，なかなかOS-9が買えません。

加藤 信夫(41)三重県

OS-9/X68000にもHuman68kのCOMMAND.Xのようなシェルがあればいいんですけどね。

◆メガドラ版のSUPER大戦略を買いました。あの内容で6800円はすっごく安い。X68000版(発売日に買った)と比べたらそれはもう……ですよ。X68000は環境がどうこう言う人もいますが，メガドラ版であれだけのことができるのだから，要はやる気だと思います。それから，T&Eソフトは X68000用ソフトをなんにも出していないくせにえらそーなことは言わないでね。

富田 祐樹(16)東京都

ゲーマーの厳しい意見。でも両方持っているなんて羨ましい。

◆こんにちは，Oh!Xは朝の電車の中で読むことが多いのですが，ちょっとした言い回し(書き回し？)がミョーに笑えるときがあります。6月号では荻窪圭氏の「備えさえあれば，幸せいっぱいの人々。」のLANに関するダジャレみたいな部分を読んでいて，思わずプププッと吹き出してしまいました。周囲の白い視線……。いま読み返してみてもどうということはないのに。これからも遊び心いっぱいの(でも中身はコイめの)記事を楽しみにしています。

川津 吉博(24)埼玉県

荻窪氏の原稿は生ものですから，リアルタイムに受けるんですよね，きっと。

◆創刊号より買い続けて以来，初めてハガキを出します。創刊当時は学生で，MZ-80Kを使い，マイクロマウスを作っていました(当然，マシン語の鬼と化していました)。近ごろは会社からの帰宅も遅く，家計簿処理機と化している我がX1です。夏にはX68000を買い，久々にまじめにプログラマしようと思います。ちなみに会社には，5550, PC, J-3100, M50などがうようよしています。

井上 勉(26)神奈川県

往年のドラゴンが復活！ なんてね。

◆ゲテモノドリンク愛好家の皆さん。私はとうとう究極のゲテモノドリンク(略してゲテドリ)を見つけました。大抵のもの(たとえば，スイカソーダ，メロウレッド，サスケなど)をこなしてきた私も，こいつだけは全部飲むことができませんでした。そいつの名は「麦コーラ」です。味のほうは，濃い麦茶に炭酸が入っていて，なおかつ甘いというとんでもないものです。皆さん，ぜひチャレンジしてください。

大橋 秀明(18)神奈川

◆山瀬まみと井森美幸がアイドルでデビューしていたなんて最近まで知りませんでした。しかし，今や立派なコメディアンの2人が口を揃えて「今もアイドルのつもりです」とゆーのにはまいった。もう誰も信じてないのに××。しかし，マックのマウスカバーにホレ込んでしまいました。シャープさんもあーゆーかわいいカバー作ったらいいのに(誰か私にください，ついでに本体も……あつかましい)。

鶴見 明子(20)東京都

そういえば，祝さんて山瀬まみのファンじゃなかったっけ……。

◆わたらせ渓谷鉄道ってご存じですか？ 群馬県の桐生市から栃木県の足尾町間を結ぶ第3セクター路線です。私は祖父母が足尾に住んでいるので，よく利用します。10年後には黒字転換を狙っているそうですが，ちょっと無理そうなので，皆様，一度は乗ってやってください。よろしくお願いします。

須永 望(16)埼玉県

◆(で)さん宛のメッセージは僕です！ 東京から1時間以上離れ，○巻温泉の奥にある，鶏を飼っている，山の上にある学校っていえばここしかないでしょう。心理学の教科書も一緒だったし。今度サインくださいね。

大村 健一(20)静岡県

話の見えない人は6月号のSHIFT BREAKを見てね。

◆友人の家で「FRハンドブック'88」という本を見つけて読んでみたら，56ページの第1回ファ

▲鶴見 明子(20)東京都 イラストだかなんだかわからないけど，編集部では思いっきり受けてしまった。で，結局どーなったの？ 今度教えてね。

▲味野 真一(23)群馬県 ソーサリアンは息の長いゲームですね。イラストのネタも尽きないようだし。ちなみに「すでに石」というのには笑ってしまった。

ンロードコンテスト，コンピュータプログラム部門選外作品紹介というところに「埼玉県　古村聡(19)」とあったんだけど，これってやっぱり(で)さんでしょ。　糟谷　輝正（16）東京都

今やすっかり人気ものとなった(で)君らしい過去ですねえ。

◆現在，昇任試験勉強中のため，X68000は近くのパソコンショップに預けてあります。そこは，PC-98，286，FM TOWNSは置いてあったのですが，Xfamilyはなかったので，喜んで展示させてくださいました。店の方はFMよりX68000のほうがよいと絶賛，店用に購入するとのことです。シャープさん，何かおくれ。

新穂　義久（32）宮崎県

宣伝力に乏しい(?)シャープですが，ユーザーの結束力はどこにも負けないね。

◆Oh!Xも消費税による4円の便乗値上げ！　うむ……，こうなったら広告をもっと増やして値段を税込で500円にしてもらいたい。広告はXシリーズに関係なくてもよい。

桧本　住文（18）大阪府

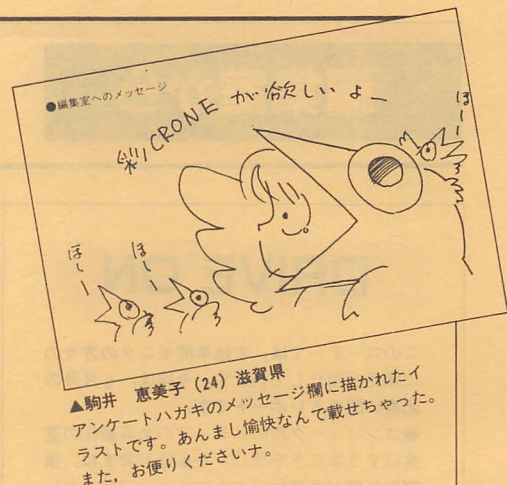
もちろん，広告が増えればOh!PCみたいに厚くても安くできるんですけど……FM TOWNSの広告でもいいですか？

◆最近のOh!Xは，宮沢賢治の「注文の多い料理店」のように感じる。石川　一彦（29）石川県

最後は，謎のメッセージ。なんとなくわからないでもないですが，皆さんはどう解釈されるでしょう。ではまた。お返事は(S.S.)でした。

▲駒井　恵美子（24）滋賀県
アンケートハガキのメッセージ欄に描かれたイラストです。あんまし愉快なんで載せちゃった。また，お便りくださいナ。

ぼくらの掲示板

●掲載ご希望の方は，官製ハガキに項目(売る・買う・氏名・年齢・連絡方法……)を明記してお申し込みください。
●ソフトの売買，交換については，いっさい掲載できません。
●取り引きについては当編集室では責任を負いかねます。
●応募者多数の場合，掲載できない場合もあります。

仲　間

★「ZW（ゼータワラビーズ）」では，MZ-700/1500ユーザーを対象とした会員を募集しています。2カ月に1度，会報を発行し，活動内容はプログラムの共同入力，Oh!Xに会報を送ってやろうじゃねえか計画などもやっています。漫画もあります。興味のある方は往復ハガキにて連絡を。〒202 東京都保谷市本町1-16-2　日本電子開発(株)柳沢寮　山田俊英（21）

★8ビットから32ビット，ホビーからビジネスまで，パソコンを使いながら楽しむクラブです。PC-9801，X68000，FMR，AX，MSX，X1など，どのパソコンユーザーでも入会できます。入会希望の人は62円切手を貼った返信封筒をお送りください。〒604 京都府京都市中京区蛸薬師境町西北角 みよいビル内　コンピュータ・ファン・クラブ

★MZ-1500，X1，S-OS他のサークル「EXTRA」では会員を募集しています。会報は2カ月に1回発行し，会費は1回117～153円，入会金は100円です。100円小為替（のみでOK）を送ってもらえればサンプルの会報と案内書を送ります。案内書だけ希望の方は62円切手を送ってください。会報の内容は，お便り，Q&A，技術的な情報，ゲーム解説，ショートプログラムなどです。投稿があれば，他機種のものでも載せますので，他機種ユーザーの方も参加してくれるとうれしいです。〒811-42 福岡県遠賀郡岡垣町戸切794-3　筑紫高宏（22）

売ります

★MZ-1P17（24ピンカラープリンタ）を，ケーブル，リボン，第2水準ROM付きで2万5千円にて。連絡は往復ハガキで，早い者勝ちです。〒739-17 広島県広島市安佐北区落合南4-41-6　小野靖弘（18）

★X1用FM音源ボードCZ-8BS1（完動，箱なし，付属品なし）を1万円前後で。連絡は往復ハガキで。〒133 東京都江戸川区谷河内町1-30-6　中村庸一（17）

★X1用FM音源ボードCZ-8BS1（付属品・箱あり）を1万1千円で送料込み。連絡は往復ハガキで。〒569 大阪府高槻市芥川町2-3-16　木下勝文（17）

★X1用FM音源ボードCZ-8BS1（ソフトあり，箱なし，スピーカあり）を5千円で。NEW Z-BASIC CZ-141SF（2D，2DD，64Kボード，箱等完全品）を8千円で。エプソンのプリンタVP-800完全美品（備品等すべてあり）を4万円くらい，またはX68000 ACEシリーズ用1MBボードCZ-6BE1Aとの交換可。連絡は往復ハガキで。〒156 東京都世田谷区羽根木1-25-8川合荘203号　川浪輝之(21)

★X1Fmodel20用増設フロッピーディスクCZ-52FE（完動，傷なし）を送料込みで8千円。連絡は往復ハガキで。〒275 千葉県習志野市東習志野2-2-4-307　千田孝之（19）

★X1用FDDインタフェイスCZ-8B01。X1用300ボーモデムCZ-8TM1。各4千円で。〒787 高知県中村市一条通3-3-16　伊藤明彦（34）

★カラーイメージボードCZ-8BV1を1万円で。プリンタMZ-1P17にケーブル，リボンを付けて2万円で。連絡は往復ハガキで。〒146 東京都大田区鵜の木1-17-9　山上勝也（29）

買います

★X1用ディスプレイテレビCZ-800D（付属品付き，色問わず）を1万～2万円で。売値を書いて往復ハガキで。〒393 長野県諏訪郡下諏訪町町屋敷2200　黒田俊一（16）

★X1用データレコーダCZ-8RL1を5千円前後，FM音源ボードCZ-8BS1を1万円以内で。どちらも多少のキズは可。ただし，完動，付属品，説明書付きのもの。箱は特にいりません。連絡は往復ハガキで。〒950 新潟県新潟市鏡が岡4-15　遠藤俊行（18）

★X1用拡張I/OポートCZ-8EPを5千円以内で。X1用データレコーダCZ-8RL1を5千円以内で。X1Fモデル20を1万円以内で(FD不良でも可)。X1用漢字ROM CZ-8BK2またはCZ-8KRを5千円以内で。連絡は，状態，希望価格を明記のうえ往復ハガキで。〒960-12 福島県福島市松川町浅川字前田15　千葉茂樹（30）

★X68000用カラーイメージユニットを3万円で。連絡は往復ハガキで。〒377 群馬県渋川市下郷1318-4　河村具成（18）

★X68000用カラーイメージユニットCZ-6VT1/BK（付属品，取り説付き）を3万～3万5千円で。連絡は官製ハガキで。〒932 富山県小矢部市野寺230-11　大盛昇（18）

★昔々，Oh!MZの表紙を飾ったシドミード「ブレードランナー」画集を4千円で。連絡は往復ハガキでお願いします。〒245 神奈川県横浜市泉区和泉町5732-2　丸藤俊之（20）

バックナンバー

★Oh!MZ1985年5～11月号を送料込み各1,000円で。切り抜き不可。連絡は往復ハガキで。〒188 東京都田無市向台町3-4-82　鬼頭良太郎（19）

★Oh!MZ1985年5月号を送料込み1,500円で。連絡は往復ハガキで。〒201 東京都狛江市和泉本町4-7-22-305　庄島賢一（20）

★6月号で「X1用FDD CZ-503Fか拡張BOXを買います」と掲載してもらったとき，売りたいと希望された方，当方の住所が変わったため，申し訳ありませんがもう一度ご連絡ください。〒180 東京都武蔵野市西久保2-15-30湖国寮　平野岳志（18）

編集室から from E・D・I・T・O・R

DRIVE ON

このコーナーでは，本誌年間モニタの方々のご意見を紹介しています。今回は，6月号の記事に関するレポートです。

●コンピュータ用のメディアの大容量化の速度はすさまじさを感じさせるほどですね。情報の形態が多様化（とくにイメージデータ）している現在，メディアの大容量化への要望も強いのでしょうが，そのとき問題になるのがファイルを取り扱う側のソフトウェアだと思うのです。現状では，UNIXに源を発する階層化ディレクトリ，Macに代表されるウィンドウシステムなどが挙げられますが，扱うファイルの数が増えれば対応が困難になってくるのは目に見えています。ハードウェアの進化とともに，それを扱うソフトウェアの研究が大切なのだろうと思います。荻窪圭氏の「備えさえあれば，幸せいっぱいの人々。」は楽しく読ませていただきました。このような世の中はすぐそこまで来ているのかもしれませんね。また，「アセンブラより小さなコンパイラ」TTCには驚かされました。ピコピコゲームなどを書くのにもいいかもしれません。ただ，やはりコンパイラに不慣れな人にはエディタとの行き来が面倒に感じられるのでは。それから，「C調言語講座 PRO-68K」で祝氏はしきりに「いーかげん」だと強調していますが，要はバランスの問題で，フライトシミュレータとしてのツボをきっちりと押さえていて，本格派目指して高速化するにも，適当に改造して派手なゲームに仕立てあげるにも，とても好都合だと思います。

今野　和浩（18）MZ-2500，FX-860PVC/780P，PB-100　埼玉県

●「学習リモコンの製作」で桒野氏がまたやってくれたので嬉しいかぎりです。設計の段階から記してくれるので，その製作過程がとても参考になりました。難しすぎることもなく，回路の構成もとてもわかりやすかったです。こういう記事を読むと確かにムズムズと動き出すものを感じますね。ちなみにTLN105AにはTPS610/611や703～706などと相性が良いようです。とくに，TPS703～706はリモートコントローラ用とのことです。あと，ジョイスティックポートの信号がどうなっているかなども記されていれば，今後読者の研究のために役立ったと思います。次回はパソコンをオシロスコープにしましょう。「X68000マシン語プログラミング」では，KILL.BATがよかったです。小さいプログラムですが，そのままユーティリティとして使えるのがいいと思います。実戦的マシン語講座の入門編らしいですね。基本的なことがすんだら大きなプログラムに進むのでしょうが，ユーティリティやコンパイラなどにも期待しています。

星　大地（16）X68000，MZ-700，PC-1475　静岡県

●「パソコンに思想と想像力を」などにも出てきた有田氏の考え方は大好きです。パソコンの場合，「いいセンスをしている」というのは何年かたったときに初めていえるものだと思うし，その間のパソコンの変化によってもセンスというものは変化すると思います。一般に，人はスペックに敏感で思想には鈍感ですから，祝氏の記事で紹介されていたようにX68000に大容量メディアが使えるようになったことは，より多くの人にX68000のよさを知ってもらう機会が増える，とも考えられるでしょうね。

橋本　浩二（18）X68000ACE-HD　兵庫県

●「備えさえあれば，幸せいっぱいの人々。」は，極近未来の人間とコンピュータのかかわりを楽しく明るく柔らかくまとめてあり，たいへん好感の持てるものだった。ひさびさのヒットですね。CPUも周辺装置も加速度的に高機能になっている現在，マンマシンインタフェイスがいかに人間寄りにあるかが問われてくるでしょう。VSはまだまだ不満がいっぱいで，早くバージョンアップしてほしいものです。「パソコンに思想と想像力を」で有田氏のいっている「センスが違う」というのは，確かにMacについて感じますね。ハード的にはX68000も確かにセンスがいいといえるでしょう。しかし，ソフトウェアをみるかぎりまだまだです。Macなどの良い点はどんどん吸収していってほしいものです。「C調言語講座PRO-68K」も少しノッてきたなと思いますが，シミュレータの解説とプログラムだけでは「言語講座」という部分をとったほうがいいのではと考えてしまいますよ。Apple社から提訴される程のことをやりたい，といっていた意気をぜひ復活させてください。

青木　民夫（34）X68000ACE-HD，PC-9801CX　富山県

●コンピュータグラフィック／ミュージックの進化もまた目覚ましいものがある。「グラフィックの可能性を探る」，「正しく“音楽する”ための基礎知識」などを読むと改めてそう感じる。しかし，ここでも性能の向上が先走ってしまい，データの互換性など肝心なことが忘れられているように思う。コンピュータで扱うデータの中でもとくに互換性が重要視されるべきデータだと思うので，ぜひ今後の対応に期待したい。そうすれば可能性はますます大きくなるだろう。

渡辺　知巳（17）X1turboZ　北海道

●「Spirit of Rally」，これはもうドライブゲームではなくてシミュレーションだ。プログラムもとても短い。より深くラリーの面白さを味わえそうだ。ギアの正しい使い方とか，ペースノートの利用法とか。一見ありがちだが，いままでにないタイプのゲームにも見える。ペースノートという存在がそうしているのだと思うが，こういうちょっとしたことでゲームのイメージがまったく変わる。うまい。

伊藤　紀之（18）X68000ACE-HD　茨城県

●本格的ドライブシミュレーションというのが似合いそうなのが「Spirit of Rally」だ。坂道の上り下りによる3次元への進化をしたとはいえ未だに指先だけの反射ゲームでしかないカーアクションゲームよりよっぽど遊べる。確かにOUT RUNは面白いしグラフィックもきれいだが，しょせんゲームの世界であり，空中回転して事故ってもまだ生きているなんて卑劣である。だから，ギアチェンジというのが実に新鮮に感じられる。リストの長さも手頃でいいと思う。「X68000マシン語プログラミング」はよかった。X68000という高性能パソコンを前にして全然複雑さを感じさせない気楽さもいい。ラベル定義などZ80のアセンブラでもやったことが出てきて入りやすかった。できればそのうち「質問コーナー」を常設してほしい。僕のようなユーザーにとってはZ80と違ってまだまだ未知の世界ですから。

上野　壮也（18）MZ-1500　大阪府

ごめんなさいのコーナー

6月号　質問箱

P.161　X1turboでBIOS内ルーチンを利用する際に割り込み禁止の指示が抜けていました。BIOSを呼び出すときは割り込みを禁止したうえで，8000_H以降に処理ルーチン，スタックを置くようにします。

5月号　戦略的ライトサイクルゲーム

P.124　X1版で座標系の変換を間違えていました。

　1190　fda=POINT(X,Y)

に修正してください。

1988年3月号　SLANG

P.124　SLANGのピリオド不等号で，負数同士の比較を行った場合の結果が誤っていました。

　$6AB6_H$　DB→DC

　$6ACB_H$　D0→D1

に修正してください。

バグに関するお問い合わせは
☎03(230)7683(直通)
月～金曜日16：00～18：00

お問い合わせは原則として，本誌のバグ情報のみに限らせていただきます。入力法，操作法などはマニュアルをよくお読みください。

また，よくアドベンチャーゲームの解答を求めるお電話をいただきますが，本誌ではいっさいお答えできません。ご了承ください。

夏です たまにはおそとで 遊びましょ

▼X1特集，いかがでしたか。新旧交代の激しいコンピュータハードウェアの世界にあって，何年たってもユーザーを失わないマシンのよさが改めて実感されたことでしょう。X1関係の投稿にも活が入ることを期待しています。

▼グラフィック特集はZバッファアルゴリズム後編と3Dモデリングの実践，サイクロンExpressのレポートなど。前回に続き読みごたえのある内容です。皆さんの感想を聞かせてください。

▼新連載「(で)のショートプロぱーてぃ」が始まりました。SHORT ACCESSみたいなのはぜひ毎月やってくれという多くの読者のリクエストにお応えして，どしどし紹介していきたいと思います。あっと，それから作品投稿のほうもよろしくね。

▼ところで，「われら電脳遊戯民」が先月最終回を迎えましたが，楽しかったので復活させてほしいとの声も届いています。今度はこんな連載がいい，などの意見がありましたら，ぜひ編集部のほうへ。

▼今回のDRIVE ONで第3期愛読者モニタの方々のレポートは最終回。モニタの皆さん，お疲れさまでした。次回からは新規メンバーです。ご期待ください。

▼今月はページの都合により，残念ながらOh!X LIVE in '89はお休みします。ミュージックファンの皆さん，ごめんなさい。ところで，最近は投稿の常連やスタッフの作品ばかりだという不満の声もあります。このページをもりあげるためにも，ぜひ，多くの読者の力の入った作品をお待ちしています。ひとつよろしく。

▼夏です。たとえ日本が世界一の財政赤字を抱えていようと，どの球団がどこに何連敗しようと，夏の休日を考えるとやはりウキウキしますよね。インディアナ・ジョーンズ3はもうみました？　遠出の予定は完璧ですか？　おっと，受験生にとってはそれどころじゃない正念場でした。食事も睡眠も十分にとって，暑さにめげずがんばってください。勉強の合い間にはOh！Xで息抜きして。お便りなんかも待ってます。

それではまた来月。

投稿応募要領

●原稿には，住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。

●プログラムを投稿される方は，詳しい内容の説明，利用法，できればフローチャート，変数表，メモリマップ（マシン語の場合）に，参考文献を明記し，プログラムをセーブしたテープ（ディスケット）を添えてお送りください。また，掲載にあたっては，編集上の都合により加筆修正させていただくことがありますのでご了承ください。

●ハードの製作などを投稿される方は，詳しい内容の説明のほかに回路図，部品表，できれば実体配線図も添えてください。編集室で検討の上，製作したハードが必要な場合はご連絡いたします。

●投稿者のモラルとして，他誌との二重投稿，他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒102 東京都千代田区九段南2-3-26井関ビル
日本ソフトバンク出版部
Oh！X「㊢㊀㋮㊔」係

S H I F T · B R E A K

▶受験は“やる気”です。極端な話，4時間パソコンしてても8時間勉強してればいいんです。封印したからって安心してごろごろしてちゃいかんよ。夏は英語。単・熟語を覚えよう。問題を解くペースが全然違ってくるぞ。現役の人は，クルマより机に向かう習慣をつけるのが先決だ。うーん，受験雑誌Oh！X。天王山みんながんばってね。（H.U.）

▶Z80ファミ・ハンを書いたのは額田忠之氏ですが，「茜さす紫野ゆき標野ゆき野守は見ずや君が袖振る」と詠んだのは額田王です。あの本を見ると，この歌を思い出します。そういえば現役受験生の冬休みは三国志を読んで終わったんだなあ。私は私立の理系だったけど。受験生諸君，休みの間に勉強するんだぞ。　P.S.すけさんもガンバるように。（S.K.）

▶私は昔から歴史書を読むということをほとんどしなかった人間だが，最近になって友人とか大学で中国語をやっているせいもあって，中国の4大奇書に大変興味をもった。今すぐにでも本屋に買いに行きたいところだけど，Oh!X8月号が発売されているころは，ちょうど前期試験の真っ最中。勉強が手につかなくなっちゃうもんね。（H.K.）

▶う゛ー，テストだ，テストだ。ついに恐怖の前期試験が始まります。さぁ，もうひとつも落とせない(で)に来年はあるか!?　というわけで来月あたり，連載開始早々スケジュール調整がとっても心配。そうそう，ちなみに私はまさか月刊アスキーにまで写真が載るとは思いませんでした。祝社長，一言いってくださいよぉ，そういうときは……。（で）

▶もう夏だ。外へと飛びだそう。おすすめは三浦半島の三戸浜。ここには旧日本軍の掘ったトンネルがそのまま放置されている。先日行ってきたのだが，たまらなく面白い。真暗闇の中では自分の目が開いているのかいないのかわからなくなるってことを実感した次第。今年の夏はアクティブにいこう。
（インディ・C・W・ジョーンズ）

▶小学校の頃，原稿用紙3枚で作文を書きなさいなんていわれると，姑息な手を使ってとにかく行数を稼いだ記憶がある。行末できっちり終わる段落は余計な読点を入れて1行増やし，“ぼく”の代わりに“わたし”を使って1文字増やすとか。最近は逆に文章が長くなりすぎて，短くするのに気を遣う。往年のテクニックが全然活かせないじゃなぁい。（Mu）

▶昔バイトで一緒だった（ついでにくどき損ねた）女の子がTV局のアナウンサーになって深夜番組に出ていた。コンビニエンスストアに友人の作ったアイドルビデオが並んでいた。OLから映画雑誌の会社にトラバーユしてしまった娘もいた。みんな，自分の時の流れを歩んでいて，自分の可能性を信じていて，見ていてとても気持ちがいい。（K）

▶出張でアメリカへ行くことになった。以前から英語はできない，外人は嫌いだ，飛行機には乗りたくないと主張していたのだが，そんなことはお構いなしに業務命令が下りてくる。会社ってこんなもんなのさ。アメリカでの技術ミーティングのことを考えると胃が痛いが，それ以外の1週間を遊んで暮らせると思えば儲けものなのかもしれない。（KO）

▶“Dead Ringers”でのクローネンバーグは少々息切れがしていた。内臓に直接響いてくるような，いつものインパクトに欠けている。一卵性双生児の違いを第3者に見せるのは，ある意味で非常に簡単だ。その違いが違いでなくなり，入れ替わりさえするようすをマスコミはかなりほめているが，伏線が平凡すぎてあくびが出た(けど面白かった)。（よ）

▶某一流電機メーカーを辞職してマンガ家になった後輩が月マガでデビューした。彼と一緒に住んでいる先輩は大学を中退してアブナイ月刊誌でマンガを描いている。代打屋トーゴーのアシスタントをやっていた先輩は元数学教師だった。薬学部を出た後輩も東京に出てマンガ家を目指している……。どうしてこうヤクザな人間が多いのだろうか。（U）

▶この8月号に掲載された「M&MII」の原稿をもって，万年ゲーマーの清水和人さんが，しばらくの間，Oh!Xから離れることになった。急な話だったので記事中にはなにも記していないが，本人いわく，「M&MIIをすべて制覇したとき,突如として私は復帰する」んだそーで，まっ，しばらくの間は，長期の充電期間だと思ってやってください。（N）

▶このままいけば来春にもX68000の出荷台数が10万台を突破する。なんのかんのいっても，個人ユーザーに10万台以上売れたシリーズは，16ビットパソコン史上，PC-9801と互換機のPC-286だけ。FMRやAXではほとんどその可能性はないし，3番手として望みがあるのは32ビットのTOWNSぐらいのものか。X68000の責任は重大だ。目指せ10万台！（T）

microOdyssey

今月の「THE SOFTOUCH」でも紹介されているが，TETRISの作者であるソビエトのゲームデザイナー，アレクセイ・バジトノフ氏が6月末から2週間ほどビー・ピー・エスの招待で来日された。運よく私も氏と直接話をする機会があったので，そのとき氏がきさくに語ってくれた，ソビエトのコンピュータ事情についてここで紹介することにしよう。

ソビエトのコンピュータ事情とひと言でいっても，ことソビエトや中国に関しては，国家体制の違いもあって公式にそういった関係のニュースはほとんど伝わって来ないのが実状だ。事実，話を聞いた私自身もごくわずかな知識しか持っておらず，最初は多少とまどっていたのだが，情況はどちらも同じだったらしく，氏からいきなり「日本では中高生くらいの若い人たちでも16ビットパソコンを持っているらしいが，彼らはいったい家でなにをするために使っているのか？」と聞かれて，その答えには苦労させられた。

基本的にソビエトでは，パーソナルユーザーというのは存在せず，日本でいうパソコンレベルのコンピュータのほとんどは，会社を中心に研究所や学校，各コンピュータセンターに設置されており，モスクワの場合であれば，コンピュータセンターに行けば，ヤマハのMSX（学校を含めて数千台入っているらしい）を始め，PDP11やIBM-PCなどのハードとともに文献も用意されており，誰でも触れることができるようになっているという。とにかく8ビット/16ビットマシンに限らず，パソコンは仕事で使うための道具であって，日本のように大部分が趣味で使うためにそれぞれの自宅に持っているなどというのは，ちょっと考えられないことだったらしい。

しかし，遊び心というのはいずこも同じらしく，氏がPDP11を使って開発したTETRISは，制作した1984年の翌年には会社を中心に大流行し，どのコンピュータのシステムにも必ずといっていいほどTETRISが入っていたという。そのため業務に支障が出始め，TETRISを立ち上げたときTETRISのデータだけを消去してしまう，対TETRISプログラムが出回ったという話まであったようだ。

コンピュータゲームに関しては，先ほどのコンピュータセンターに行けばいくつかゲームソフトも用意されているらしいのだが，日本のものとはかけ離れたレベルのゲームしか用意されておらず，最近，ゲームウォッチクラスのものが市販されたときは，店頭に行列ができるほどの人気だったらしい。このように，中学生以上には，ゲームに限らずパソコンに対する人気は高いようで，学校の授業で使うだけで満足しない学生たちは，せっせとコンピュータセンターに通って，自分たちで資料を見ながら熱心に勉強しているという。

ただ，モスクワでこのような状況なのだから，それ以外の都市も含めて考えると，マシンの台数不足は隠せないものがあるようだ。

今回，氏が来日できた背景などについては，細かいところまでは聞けなかったが，これを機に，今後コンピュータを通しての交流が深められそうな気配が感じられたのは，我々にとっては期待できる材料であるに違いない。　（N）

1989年9月号8月18日（金）発売

特集　活用 ハードディスク&プリンタ
・各社ハードディスク接続総チェック
・24ピンプリンタエミュレータ
入門：サイバースティック用プログラミング
Oh!X LIVE in'89
X68000：サンダークロス／X1：イタリア協奏曲/代々木ゼミナール校歌 他

バックナンバー常備店

東京	神保町	三省堂神田本店5F	03(233)3312
	〃	書泉ブックマートBI	03(294)0011
	〃	書泉グランデ5F	03(295)0011
	秋葉原	T-ZONE 7Fブックゾーン	03(257)2660
	八重洲	八重洲ブックセンター3F	03(281)1811
	新宿	紀伊国屋書店本店	03(354)0131
	高田馬場	未来堂書店	03(200)9185
	渋谷	大盛堂書店	03(463)0511
	池袋	リブロ池袋店	03(981)0111
	〃	西武百貨店9F コンピュータ・フォーラム	03(981)0111
神奈川	横浜	有隣堂横浜駅西口店	045(311)6265
	〃	有隣堂ルミネ店	045(453)0811
	藤沢	有隣堂藤沢店	0466(26)1411
神奈川	厚木	有隣堂厚木店	0462(23)4111
	平塚	文教堂四の宮店	0463(54)2880
千葉	柏	新星堂カルチェ5	0471(64)8551
	船橋	リブロ船橋店	0474(25)0111
	〃	芳林堂書店津田沼店	0474(78)3737
	千葉	多田屋千葉セントラルプラザ店	0472(24)1333
埼玉	川越	黒田書店	0492(25)3138
	川口	岩渕書店	0482(52)2190
茨城	水戸	川又書店駅前店	0292(31)0102
大阪	北区	旭屋書店本店	06(313)1191
	都島区	駸々堂京橋店	06(353)2413
京都	中京区	オーム社書店	075(221)0280
愛知	名古屋	三省堂名古屋店	052(562)0077
	〃	パソコンΣ上前津店	052(251)8334
	刈谷	三洋堂書店刈谷店	0566(24)1134
長野	飯田	平安堂飯田店	0265(24)4545
北海道	室蘭	室蘭工業大学生協	0143(44)6060

定期購読のお知らせ

Oh!Xの定期購読をご希望の方は，どじ込みの振替用紙の「申込書」欄に何年何月号からをご記入のうえ，年間購読料6,720円（税込）を添えてお申し込みください。その際，裏面の通信欄に「○年○月号よりOh!X定期購読希望」と忘れずに明記してください。なお，すでに定期購読をご利用いただいている方には，購読期限終了と同時にご通知申し上げますので，同封の払込用紙をご利用ください。

海外送付ご希望の方へ

本誌の海外発送代理店，日本IPS（株）にお申し込みください。なお，購読料金は郵送方法，地域によって異なりますので，下記宛必ずお問い合わせください。

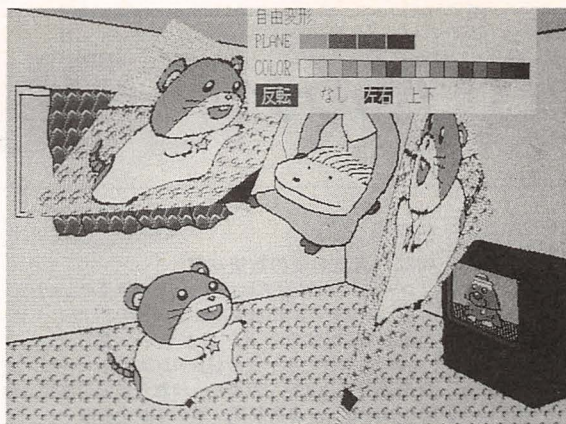
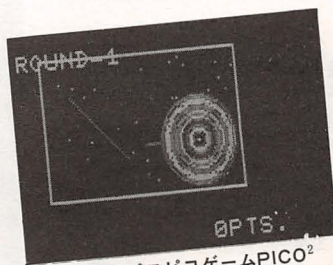
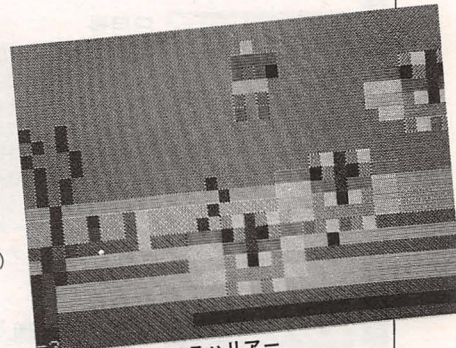
日本IPS株式会社
〒101 東京都千代田区飯田橋3-11-6
☎03(238)0700

Oh!X　8月号
■1989年8月1日発行　定価560円（本体544円）
■発行人　孫　正義
■編集人　橋本五郎
■発売元　（株）日本ソフトバンク
■出版事業部　〒102 東京都千代田区九段南2-3-26　井関ビル
Oh！X編集部 ☎03(230)7681
出版営業部 ☎03(230)7670 FAX 03(262)8397
広告営業部 ☎03(230)7672
■印　刷　凸版印刷株式会社

投稿プログラム大募集
のお知らせ

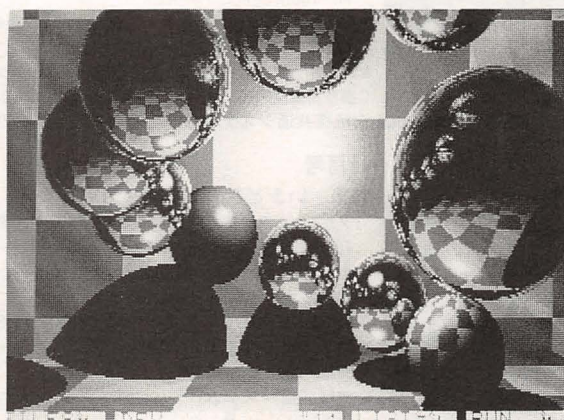
Oh!Xでは，毎月さまざまな投稿プログラムを掲載しております。これらはすべて，ゲーム音楽を聞いているうちに自分のマシンで演奏してみたくなった，市販のものもあるけどもっと便利なグラフィックツールが欲しかった，またはMZ-700でスペースハリアーを遊びたいなど，どれも皆さんが日常のなかでパソコンと接しているうちに，ふと思いついたことを形にしようと努力して生み出された傑作，名作ばかりなのです。

でも，読者の皆さんがそうして作り上げたプログラムを，一部の方を除いては自分のディスクのなかだけにしまっておくのはもったいない話。ひとりでも多くのユーザーに使ってもらえば，またそれをベースにして新しいプログラムが生まれる可能性だって広がるのです。

ですから，Oh!Xではそういったちょっとしたきっかけを機に，完成度の高いものよりも自分のアイデアをそのまま形にしたような，オリジナリティあふれる投稿プログラムをスペースを空けてお待ちしています。もちろん，ピコピコゲームのようなショートプログラムも大歓迎。自信作をお持ちの方は，募集要項をよくお読みのうえぜひご参加ください。お待ちしています。

MZ-2500用グラフィックツールDMACS（1988年9月号）

MZ-2500用ピコピコゲームPICO²
（1988年4月号）

MZ-700用スペースハリアー
（1988年10月号）

X1/X1 turbo用割り込みミュージックシステムPSI
（1988年3月号）

X68000用ストラテジーゲームSTAR TREK
（1988年11月号）

S-OS“SWORD”用ELFES
（1988年2月号）

X1turbo用レイトレーシングツールturbo RAY TRACER
（1988年9月号）

投稿募集要項

1） お送りいただくプログラムには，住所・氏名・年齢・職業・連絡先電話番号・機種名・使用言語・必要な周辺機器・マイコン歴等を明記のうえ，封書の宛て先の最後には「Oh!X LIVE」や「S-OS“SWORD”」，「投稿ゲームプログラム」など，プログラムの内容を明確にご記入ください。

2） 投稿されるプログラムには，詳しい内容を記入した原稿と一緒にフローチャート，変数表，メモリマップ，参考文献などの資料もお書き添えのうえお送りください。また，お送りいただいた原稿については，当方で加筆，修正させていただく場合があります。

3） お送りいただくプログラムは最低2回はセーブしてください。基本的に同封されたカセットテープおよびフロッピーディスクについてはご返送いたしませんので，あらかじめご了承ください。

4） ハード製作関係の投稿につきましては，最初は詳しい内容のわかる原稿のみお送りいただければ結構です。その後，当方において製作物が必要だと判断した場合は，改めてご連絡いたします。

5） お送りいただいた投稿プログラムの採用につきましては，掲載月号が決定した時点で当方よりご連絡を差し上げます。特に各種ツール関係，ハード関係のものにつきましては，特集内容などを考慮したうえで採用が決定されることがありますので，採用結果をご連絡するまでに時間がかかってしまう場合もあります。

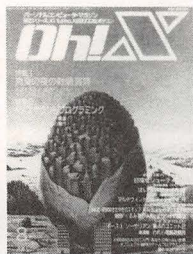
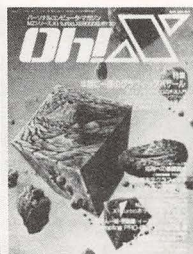
6） 投稿いただいたプログラムにバグ等が発見された場合には，新しいプログラムの入ったメディアと一緒に，文書にてご連絡ください。

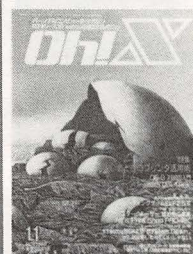
7） 掲載された投稿プログラムに対しては当社規定の原稿料をお支払いいたします。また，プログラムの著作権等は制作された方に保留されますが，PDSとしてネットなどにアップロードされる場合は，必ず編集室まで事前にご連絡ください。なお，一般的モラルとして，他誌との二重投稿または，他誌に掲載されたプログラムの移植などについては固くお断わりいたします。

宛て先
〒102 東京都千代田区九段南2-3-26 井関ビル
日本ソフトバンク Oh!X編集室「投稿プログラム」係

BACK ISSUES バックナンバー案内

ここには1988年8月号から1989年7月号までをご紹介しました。現在1987年4,1988年1,2,4,5,6,7,8,9,10,11,12,1989年1,2,3,4,5,6,7月号までの在庫がございます。バックナンバーおよび定期購読のお申し込み方法については本文176ページを参照してください。

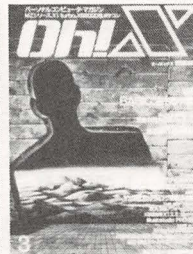
1988

8月号
特集1 真夏の夜の数値演算
コンピュータの数値表現/応用グラフィック歪められた光/AD PCM音の数学/数値演算プロセッサ用ドライバ 他
特集2 MIDIサウンドプログラミング
MIDIの基礎とボードの製作/MIDI対応シーケンサ
THE SOFTOUCH 新連載 われら電脳遊戯民 他
猫とコンピュータ第26回 ボクはかぐや姫?
新連載 Z80マシン語ゲーム工房
全機種共通システム マルチウィンドウエディタWINER

9月号
特集 半期に一度のグラフィックバザール
CGアニメの手法入門/ワイヤフレームによる3D/X68000スプライト/画像処理の基礎知識/turbo RAY TRACER/MZ-2500用グラフィックエディタDMACS
THE SOFTOUCH C-TRACE68/SAMPLING PRO-68K 他
C調言語講座PRO-68K(3) 謎の低次元グラフィック
MIDI活用テクニック(2) 割り込みによるMIDI通信
Z80マシン語ゲーム工房(2) 応用への基礎固め
全機種共通システム ラインエディタTED-750/WINERの拡張

10月号
特集 百花繚乱ゲームバトルロイヤル
最新ゲーム総登場 ハイドライド3/A列車で行こうII/たんば/熱血高校ドッジボール部/フルスロットル 他
MZ-700用SPACE HARRIER
● Oh!X LIVE 1974(16光年の訪問者)/瑠璃色の地球/二人のゼネレーション/バッハのアリア
MIDI活用テクニック(3)複数の音源を操るテクニック
C調言語講座PRO-68K(4)/Z80マシン語ゲーム工房(3)
全機種共通システム SLANG用拡張ライブラリ/MANKAI

11月号
特集 いまどきのプリンタ活用術
メカニズムを理解しよう/制御コード/文字と図形の混在印字/拡大文字のスムージング/外字登録ツール/S-H COPY/グラフィックのモノクロ出力/X68000のCOPYキー/オリジナル印刷キット/試用レポート
THE SOFTOUCH NEW Print Shop PRO-68K 他
OS-9/X68000入門(1) OS-9ってなに?
● STAR TREK for X68000
全機種共通システム シューティングゲームELFES IV

12月号
特集 パソコンはいま音楽の領域へ
なぜ自動作曲か/心地よい雑音の話/和音の読み方/美しい響きの要素/4分音符は歌い始める/古くて新しい音楽形式/FM音源の仕組み/Melody Box/MusicBASIC
● さよなら Live in '88 バッハ イタリア組曲他6本
● Oh!X 1周年記念特別企画「ちょっとあぶない福袋」
OS-9/X68000入門(2) OS-9のオペレーション環境
Z80マシン語ゲーム工房/C調言語講座PRO-68K
全機種共通システム ソースジェネレータSOURCERY

1989

1月号
特集 いきなり初春からハードウェア
デジタル回路入門/電子サイコロ/乱数発生器/X1turboバンクメモリ拡張/X68000用CP/M-80システム 他
1988年度GAME OF THE YEAR ノミネート作品発表
● MZ-2500用 Hyper Game Book
● LIVE in'89 エンデューローレーサー/アルルの女
● ようこそ,セガ・メガドライブ!!
C調言語講座PRO-68K/Z80マシン語ゲーム工房
全機種共通システム パズルゲーム LAST ONE/FLICK

2月号
特集 マシン語“でじたるざんまい”
アーキテクチャからのマシン語入門/アセンブラへの招待/超入門Z80マシン語活用術/X68000料理教室
THE SOFTOUCH 彩CRONE/Final Ver.3.2 他
● X1/X1turbo用RPG FLAME
Z80マシン語ゲーム工房 最終回 爆発,そして完成へ
C調言語講座PRO-68K (8) とおりゃんせなのである
OS-9/X68000入門(3) ついに発売! OS-9/X68000
全機種共通システム 高速エディタアセンブラREDA

3月号
特集 BASIC“おもちゃ箱”
ピコピコゲームから重力シミュレーションまで
● X1/X1turboでMZ-700用スペハリ/ロボットゲームTAMA
● 数値演算を高速化 FLOAT2+.X
OS-9/X68000入門(4) C言語の概要を見る
C調言語講座PRO-68K(9) ニホン語,不得意
新連載予告編X68000マシン語プログラミング入門
全機種共通システム 浮動小数点演算パッケージSOROBAN
THE SOFTOUCH/LIVE in'89/知能機械概論/猫とコンピュータ

4月号
特集 ゲーマーたちの“新深夜族”宣言
1988年度GAME OF THE YEAR
新連載 X68000マシン語プログラミング
● X1/turboパズルゲーム ロボット衛兵
● MZ-700用ゲームパッケージ System-7B
● LIVE グラディウスII/ザ・スキーム/パワードリフト
連載 C調言語講座PRO-68K/OS-9/X68000入門
全機種共通システム SLANG用実数演算ライブラリ
特別付録 X68000イメージCGポスター

5月号
特集 MIDIサウンドデータ料理術
LA音源をFM音源でシミュレート/X-BASICでMIDI制御
特別企画 第4回「言わせてくれなくちゃだワ」
● シャープパソコンフォーラム'89 in赤坂
● 詳解Human68k ver.2.0
● MZ-2500, X1/X1turbo用 戦略的ライトサイクルゲーム
連載 C調言語講座PRO-68K/ OS-9/X68000入門
X68000マシン語プログラミング
全機種共通システム ソースジェネレータRING

6月号
特集 これからのXfamily
X68000に光磁気ディスクを/学習リモコンの製作
THE SOFTOUCH ライトニングバッカス/Might and MagicII他
● OPMA用外部関数による KENBAN.BAS
● X1/X1turbo用ドライブゲーム Spirit of Rally
● X1turboZ用 これ,パズルなんですか。
MZ-2500 MIDI入門(1)MIDIボードを作る
C調言語講座PRO-68K/X68000マシン語プログラミング
全機種共通システム 超小型コンパイラTTC

7月号
特集 3Dグラフィックへの飛翔
Zバッファアルゴリズム/スムースシェイディング他
THE SOFTOUCH Terazzo PRO-68K/アドヴァンスト・ファンタジアン
新連載 DōGA・CGアニメーション講座
MZ-2500用グラフィックエディタ作成講座
マシン語カクテル in Z80's Bar
X-BASICプログラミング調理実習
全機種共通システム TTC用パズルゲームTIC BAN
X68000マシン語プログラミング/C調言語講座PRO-68K 他

日本ソフトバンクの
書籍特約書店

下記の書店の一覧は、日本ソフトバンク書籍特約店として右にある商品の他、新刊もとりそろえております。ご希望の商品がある場合は、下記のお近くの書店にてお買い求め下さい。
(注)現品が売れて補充中の場合もございますので、ご注意下さい。

SOFT BANK

日本ソフトバンク出版事業部

〒102 東京都千代田区九段南2-3-26 ✆03(261)4095

全国特約書店一覧

地域	書店名	電話
〈北海道〉		
札幌市	紀伊國屋書店札幌店	011-231-2131
〃	旭屋書店札幌店	011-241-3007
〃	丸善札幌支店	011-241-7252
〃	リーブルなにわ	011-221-3800
〃	冨貴堂札幌パルコ店	011-214-2303
〃	ダイヤ書房本店	011-712-2541
〃	ダイヤ書房西店	011-665-6223
旭川市	旭川冨貴堂	0166-26-3481
〃	ブックス平和マルカツ店	0166-23-6211
苫小牧市	旭屋書店苫小牧店	0144-36-5185
〈東北〉		
青森市	成田本店	0177-23-2431
〃	岡田書店	0177-23-1381
弘前市	紀伊國屋書店弘前店	0172-36-4511
〃	ブックイン城東	0172-28-2882
八戸市	伊吉書院	0178-44-1917
盛岡市	東山堂書店本店	0196-53-6464
〃	さわや書店	0196-53-4411
〃	第一書店	0196-53-3355
仙台市	金港堂	022-225-6521
〃	金港堂ブックセンター	022-223-0979
〃	アイエ書店駅前店	022-264-0718
〃	丸善仙台支店	022-266-1127
〃	髙山書店	022-263-1511
〃	ブックスみやぎ	022-267-4422
秋田市	三浦書店	0188-33-8131
山形市	八文字屋	0236-22-2150
福島市	岩瀬書店コルニエツタヤ店	0245-21-2101
〃	博向堂	0245-21-1161
郡山市	東北書店	0249-32-0379
いわき市	ヤマニ書房本店	0246-23-3481
〃	鹿島ブックセンター	0246-28-2222
会津若松市	宝文館	0242-27-5198
原町市	文芸堂	0244-22-1720
〈関東〉		
水戸市	川又書店駅前店	0292-31-0102
〃	ツルヤブックセンター	0292-25-2711
勝田市	武石書店	0292-73-1212
東海村	大野書店	0292-82-2098
鹿島郡	なみき書店	0299-96-1855
土浦市	共栄堂	0298-21-6134
つくば市	丸善筑波大学会館店	0298-51-6000
〃	友朋堂吾妻本店	0298-52-3665
宇都宮市	落合書店ホリオン店	0286-34-3777
〃	落合書店東武ブックセンター	0286-34-8271
〃	新星堂宇都宮店	0286-33-2337
小山市	進駸堂駅ビル店	0285-25-1522
前橋市	煥乎堂	0272-23-1211
〃	リブロ前橋店	0272-34-1011
〃	戸田書店前橋店	0272-61-5063
高崎市	学陽書房	0273-23-4055
〃	サカ井書店	0273-62-1500
〃	新星堂高崎店	0273-27-3961
〃	戸田書店高崎店	0273-63-5110
太田市	ナカムラヤ	0276-22-2001
〈首都圏〉		
浦和市	須原屋本店	0488-22-5321
浦和市	須原屋コルソ店	0488-24-5321
大宮市	押田謙文堂	0486-41-3141
〃	ブックセンター押田	0486-47-3141
〃	三省堂ブックポート	0486-46-2600
蕨市	須原屋蕨店	0484-44-1211
川口市	岩渕書店川口店	0482-52-2190
川越市	黒田書店川越店	0492-25-3138
所沢市	芳林堂所沢店	0429-25-5355
〃	いけだ書店所沢店	0429-28-3271
上福岡市	黒田書店上福岡店	0492-66-0120
朝霞市	文教堂朝霞店	0484-76-0107
志木市	新星堂志木店	0484-74-0182
春日部市	文教堂春日部店	0487-52-7666
比企郡	錦電サービス	0492-96-2962
千葉市	多田屋セントラルプラザ店	0472-24-1333
〃	キディランド千葉店	0472-25-2011
習志野市	巌翠堂	0474-72-5011
船橋市	ときわ書房本店	0474-24-0750
〃	リブロ船橋店	0474-25-0111
〃	旭屋書店船橋店	0474-24-7331
〃	芳林堂津田沼店	0474-78-3737
〃	第二巌翠堂	0474-65-0926
柏市	西口アサノ	0471-44-2111
〃	新星堂柏店	0471-64-8551
松戸市	堀江良文堂	0473-65-5121
〃	辰正堂駅ビル店	0473-64-7997
横浜市	有隣堂トーヨー店	045-311-6265
〃	有隣堂東口ルミネ店	045-453-0811
〃	栄松堂相鉄ジョイナス店	045-321-6831
〃	そごうブックセンター	045-465-2111
〃	丸善ブックメイツポルタ店	045-453-6811
〃	有隣堂伊勢佐木店	045-261-1231
〃	有隣堂戸塚店	045-881-2661
〃	文華堂戸塚店	045-864-5151
〃	アーバン文華堂	045-821-5151
〃	文教堂青葉台南口店	045-983-5150
川崎市	有隣堂アゼリア店	044-245-1231
〃	有隣堂川崎BE店	044-200-6831
〃	文学堂本店	044-244-1251
〃	ブックセンター文教堂	044-811-5557
〃	文教堂溝ノ口店	044-811-8258
鎌倉市	島森書店大船店	0467-46-3841
〃	鎌倉書店	0467-46-2619
横須賀市	平坂書房WALK店	0468-25-5537
藤沢市	有隣堂藤沢店	0466-26-1411
〃	リブロ藤沢店	0466-27-0111
〃	文教堂六会店	0466-82-9610
茅ヶ崎市	川上書店ルミネ店	0467-87-3827
平塚市	サクラ書店駅ビル店	0463-23-2751
〃	文教堂四之宮店	0463-54-2880
小田原市	八小堂書店	0465-22-7111
〃	伊勢治書店	0465-22-1366
〃	文教堂小田原店	0465-36-3677
厚木市	有隣堂厚木店	0462·23-4111
大和市	文教堂中央林間店	0462-75-4165
相模原市	文教堂相模大野店	0427-49-0650
〃	文教堂橋本店	0427-74-5581
〃	文教堂星ヶ丘店	0427-58-6121
津久井郡	文教堂城山店	0427-82-9278
〈東京〉		
千代田区	三省堂書店神田本店	03-233-3312
〃	書泉グランデ	03-295-0011
〃	東京堂書店	03-291-5181
〃	旭屋書店水道橋店	03-294-3781
〃	丸善お茶の水店	03-295-5581
〃	巌翠堂	03-291-1362
〃	いずみ神田南口店	03-254-8521
〃	明正堂秋葉原店	03-257-0758
中央区	八重洲ブックセンター	03-281-1811
〃	日本橋丸善	03-272-7211
〃	旭屋書店銀座店	03-573-4936
港区	書原新橋店	03-591-8738
〃	雄峰堂NS店	03-503-6586
〃	虎ノ門書房本店	03-502-3461
〃	虎ノ門書房田町店	03-454-2571
品川区	芳林堂大井町店	03-474-4946
〃	明屋書店五反田店	03-492-3881
渋谷区	紀伊國屋書店渋谷店	03-463-3241
〃	旭屋書店渋谷店	03-476-3971
〃	三省堂書店渋谷店	03-407-4545
〃	大盛堂書店	03-463-0511
〃	紀伊國屋書店笹塚店	03-485-0131
新宿区	紀伊國屋書店本店	03-354-0131
〃	三省堂書店新宿西口店	03-343-4871
〃	福家書店センタービル店	03-345-1246
〃	福家書店野村ビル店	03-342-0298
〃	新星堂NSビル店	03-344-2055
〃	西武新宿ブックセンター	03-208-0380
〃	芳林堂高田馬場店	03-208-0241
〃	未来堂	03-200-9185
豊島区	旭屋書店池袋店	03-986-0311
〃	芳林堂池袋店	03-984-1101
〃	リブロ池袋店	03-981-0111
〃	三省堂書店池袋店	03-987-0511
〃	新栄堂本店	03-984-2345
〃	新栄堂アルパ店	03-988-0181
台東区	明正堂中通り店	03-831-0191
墨田区	リブロ錦糸町店	03-846-0111
〃	ブックストア・談	03-635-1841
江戸川区	文教堂西葛西店	03-689-3621
大田区	アクトブックスサンカマタ店	03-735-1551
〃	竜文堂大森駅ビル店	03-775-3851
中野区	明屋書店東京本社	03-387-8451
杉並区	ブックセンター荻窪	03-393-5571
〃	書原杉並店	03-313-4778
武蔵野市	紀伊國屋書店吉祥寺東急店	0422-21-5543
〃	弘栄堂吉祥寺店	0422-22-1031
〃	パルコブックセンター吉祥寺	0422-21-8122
調布市	真光書店	0424-87-2222
府中市	啓文堂	0423-66-3151
三鷹市	三省堂書店三鷹店	0422-48-4510
〃	東西書房	0422-46-0275
小金井市	文教堂小金井店	0423-86-0161
国分寺市	三成堂国分寺店	0423-25-3211
国立市	東西書店	0425-75-5061
小平市	文教堂小平店	0423-43-9229

展示図書一覧

書名	価格
MS-DOSいたれりつくせり本	●1800円
プレイMS-DOS	●1900円
UNIX System V プログラマ・ガイド	●12000円
UNIX System V ユーザ・ガイド	●9800円
UNIXオペレーティングガイド	●3000円
C言語の活用理解	●2000円
C言語の基礎知識	●2500円
C言語の応用50例	●2300円
上級・C言語の応用例50例	●2400円
Cプリプロセッサ・パワー	●2200円
Play the C 上巻	●1500円
Play the C 下巻	●1500円
Turbo C入門	●2600円
8086アセンブリ言語	●2800円
8086マクロプログラミング	●2600円
ビギニングMUMPS	●2600円
Final Ver.4.0ブック	●2400円
MIFES Ver.4.0ブック	●2400円
ビジネスソフトデータ活用ブック	●2800円
BASICによるプログラミングスタイルブック	●1800円
ソーティング・ノート	●1900円
BASICプログラムジェネレータ集	●2800円
98/88スモールビジネスプログラム集	●2500円
88デスクアクセサリ集	●2000円
J-3100パワーユーザーブック	●2400円
フロッピーディスクフル活用ガイド	●2300円
PC工作入門	●1800円
続・PC工作入門	●1800円
PC-286Lブック	●1700円
試験に出るX1	●2800円
Lotus1-2-3ガイドII	●2500円
MS-Chart Ver.3.1ガイド	●2900円
まいと～くガイド	●2300円
新松ガイド	●2000円
一太郎Ver.3ガイド	●2500円
新一太郎ガイド	●2300円
桐Ver.2ガイド	●2500円
花子応用ガイド	●2500円
Lotus 1-2-3ガイド	●2400円
P1ガイド	●2300円
Ninja2ガイド	●2300円
Multiplan Ver.3.1ガイド	●2400円
アセンブラCASL入門	●2000円
ハードウェア徹底マスター	●2500円
FORTRAN徹底マスター	●2800円
情報処理の基礎知識	●1600円
COBOL徹底マスター	●2900円
受験用語ハンドブック	●1800円
ワープロ文書F・O・P	●1200円
バイト&ワードの風にのって	●1800円
田原総一朗のパソコンウォーズ	●1400円
コミック・トロン革命	●1200円
ムーグ・ノイマン・バッハ	●1300円
RPG幻想事典	●1500円
RPG幻想事典・日本編	●1800円
魔法王国シムルグント	●1800円

地域	書店	電話
東村山市	文教堂東村山店	0423-96-1115
立川市	オリオン書房ウイル店	0425-27-2311
八王子市	くまざわ書店本店	0426-25-1201
町田市	有隣堂町田店	0427-23-3018
〃	久美堂本店	0427-25-1330
〃	久美堂小田急店	0427-27-1111
〃	久美堂東急ハンズ店	0427-28-2772
〃	文教堂鶴川店	0427-35-4117
〃	文教堂小川店	0427-96-1781
多摩市	くまざわ書店桜ヶ丘店	0423-37-2531
福生市	文教堂福生店	0425-53-7708
〈甲信越・北陸〉		
甲府市	文教堂甲府店	0552-22-4600
長野市	平安堂長野店	0262-26-4545
〃	長谷川書店	0262-26-2122
上田市	平安堂上田店	0268-22-4545
松本市	ブックスロクサン	0263-35-5555
〃	改造社松本駅ビル店	0263-36-3777
飯田市	平安堂飯田店	0265-24-4545
岡谷市	笠原書店	0266-23-5070
諏訪郡	平安堂下諏訪店	0266-28-1111
新潟市	紀伊國屋書店新潟店	025-241-5281
〃	萬松堂	025-229-2221
〃	北光社	025-228-2321
長岡市	覚張書店	0258-32-1139
〃	ブックセンター長岡	0258-36-1360
〃	長岡技大長峰文化	0258-46-6437
山北町	BOOKメディア	0254-77-3850
富山市	瀬川書店	0764-24-4566
〃	清明堂	0764-24-4166
〃	BOOKSなかだ豊田店	0764-32-1353
〃	文苑堂本郷店	0764-22-0552
〃	文苑堂赤江店	0764-33-0321
高岡市	文苑堂	0766-21-0333
〃	文苑堂横田店	0766-21-0431
金沢市	うつのみや片町店	0762-21-6136
〃	書林香林坊本店	0762-20-5011
野々市町	王様の本本店	0762-46-5325
福井市	勝木書店	0776-24-0428
〃	品川書店新田塚店	0776-24-1112
〈東　海〉		
静岡市	静岡谷島屋呉服町本店	0542-54-1301
〃	江崎書店	0542-54-4481
〃	吉見書店	0542-52-0157
〃	戸田書店ＳＢＳ店	0542-81-5733
〃	戸田書店曲金店	0542-81-5899
沼津市	吉野屋	0559-23-5676
〃	マルサン書店宝塚店	0559-63-0350
富士市	戸田書店富士店	0545-51-5121
清水市	戸田書店本店	0543-65-2345
浜松市	浜松谷島屋連尺本店	0534-53-9121
名古屋市	三省堂書店名古屋店	052-562-0077
〃	星野書店近鉄ビル店	052-581-4796
〃	丸善名古屋支店	052-261-2251
〃	丸善ブックメイツセントラルパーク	052-971-1231
〃	日進堂上前津店	052-263-0550
〃	三洋堂パソコンショップΣ	052-251-8334
〃	三洋堂いりなか本店	052-832-8202
名古屋市	ちくさ正文館本店	052-741-1137
〃	白樺書房西店	052-774-7223
豊橋市	精文館	0532-54-2345
岡崎市	ブックス鎌倉	0564-54-1822
豊田市	三洋堂梅坪店	0565-35-2334
刈谷市	三洋堂刈谷店	0566-24-1134
春日井市	三洋堂勝川店	0568-32-7806
岐阜市	自由書房	0582-65-4301
大垣市	大洞堂ブックス258	0584-81-2553
〃	大洞堂岐大バイパス店	0584-74-7766
一宮市	三洋堂一宮店	0586-77-5734
可児市	三洋堂可児店	0574-63-2334
多治見市	三洋堂多治見店	0572-24-0340
津市	別所書店11ビル店	0592-24-1014
四日市市	文化センター白揚	0593-51-0711
鈴鹿市	シェトワ白揚スズカ	0593-82-5221
〈近　畿〉		
京都市	駸々堂京宝店	075-223-1003
〃	アバンティ・ブックセンター	075-682-5031
〃	オーム社書店河原町店	075-221-0280
〃	ジュンク堂京都店	075-252-0101
〃	オーム社書店竹田店	075-644-2611
奈良市	駸々堂大丸店	0742-26-6241
大阪市	旭屋書店本店	06-313-1191
〃	紀伊國屋書店梅田店	06-372-5821
〃	オーム社書店大阪店	06-345-0641
〃	駸々堂京橋店	06-353-3209
〃	駸々堂心斎橋店	06-251-0881
〃	旭屋書店ナンバ店	06-644-2551
〃	ナンバブックセンター	06-644-5501
〃	ヒバリヤ書店ナンバ店	06-644-5407
〃	旭屋書店アベノ店	06-631-6051
〃	ユーゴー書店	06-623-2341
〃	河村書店	06-951-2968
枚方市	水嶋書房京阪デパート店	0720-51-3432
高槻市	コーベブックス西武高槻店	0726-83-1766
東大阪市	ヒバリヤ書店本社	06-722-1121
神戸市	ジュンク堂センター街店	078-392-1001
〃	ジュンク堂サンパル店	078-252-0777
〃	海文堂書店	078-331-6501
〃	日東館書林	078-391-8701
姫路市	新興書房	0792-85-3344
〃	誠心堂書店	0792-81-2055
和歌山市	宮井平安堂	0734-31-1331
〃	帯伊書店	0734-22-0441
〈中　国〉		
岡山市	紀伊國屋書店岡山店	0862-32-3411
〃	丸善岡山支店	0862-31-2261
津山市	津山ブックセンター	08682-6-4047
広島市	紀伊國屋書店広島店	082-225-3232
〃	丸善広島支店	082-247-2251
〃	金正堂	082-248-3715
〃	積善館	082-248-3151
尾道市	啓文社尾道店	0848-37-5151
福山市	啓文社福山店	0849-22-3111
〃	ブックシティ啓文社	0849-25-0050
〃	啓文社コア	0849-41-0909
山口市	五十部誠文堂	0839-24-6630
山口市	文栄堂	0839-22-5611
下関市	中野書店	0832-22-6181
宇部市	京屋書店	0836-31-2323
〃	末広書店	0836-31-0086
防府市	誠文堂国衙店	0835-25-1988
光市	三文字屋	0833-71-0251
鳥取市	富士書店	0857-23-7271
松江市	園山書店	0852-21-4167
〈四　国〉		
徳島市	小山助学館本店	0886-54-2135
〃	小山助学館東口店	0886-25-1380
〃	森住丸善	0886-23-3228
高松市	宮脇書店本店	0878-51-3733
丸亀市	宮脇書店丸亀店	0877-22-5533
松山市	紀伊國屋書店松山店	0899-32-0005
〃	明屋書店本店	0899-41-4141
〃	明屋書店大街道店	0899-41-4242
〃	丸三書店	0899-31-8501
新居浜市	明屋星原店	0897-44-4000
宇和島市	明屋宇和島店	0895-23-1118
高知市	金高堂	0888-22-0161
〈九州・沖縄〉		
福岡市	紀伊國屋書店福岡店	092-721-7755
〃	りーぶる天神	092-713-1001
〃	積文館新天町店	092-781-2991
〃	福岡金文堂本店	029-741-2106
〃	福岡金文堂朝日ビル店	092-431-1094
〃	福岡金文堂デイトス店	092-451-6175
〃	福岡金文堂アニマート原	092-844-0088
北九州市	ナガリ書店	093-521-1044
〃	金栄堂	093-531-3685
〃	旭屋書店北九州店	093-631-6421
〃	井筒屋ブックセンター	093-641-0131
〃	カルパーク平野	093-661-7988
〃	白石書店本城店	093-601-2200
久留米市	エマックスたがみ	0942-33-1841
飯塚市	BOOKリード	0948-25-7266
大分市	パルコブックセンター大分店	0975-35-0643
〃	本町晃星堂	0975-33-0231
別府市	明林堂	0977-23-2183
宮崎市	中央、田中書店	0985-24-3511
〃	寿屋宮崎店	0985-27-4111
佐賀市	金華堂北バイパス店	0952-32-1965
〃	積文館デイトス店	0952-23-7155
長崎市	メトロ書店	0958-21-5453
〃	好文堂	0958-23-7171
佐世保市	金明堂書店	0956-22-4214
熊本市	紀伊國屋書店熊本店	0963-22-5531
〃	長崎書店	0963-53-0555
人吉市	明屋人吉店	0966-22-5486
鹿児島市	春苑堂ブックプラザ	0992-25-3200
〃	ブックスみすみ	0992-57-1011
那覇市	球陽堂書房ビル店	0988-63-3752
〃	文教図書	0988-62-1201

外国製のMS-DOSにもアクセス出来る!

新発売

X68000用

SUPER DEVICE MONITOR "T"

X1turboや、MZ-2500ではもうお馴染の『SUPER DEVICE MONITOR "T"』のX68000用がいよいよ発売されます。

今までは、手探りで行なっていたプログラムの開発が、容易に出来る様に成ります。
例えばCコンパイラーや機械語を使ってソフトを自作している場合、1バイトの定数等を書き換えるのにいちいちエディターでソースプログラムを書き直してからアセンブルをし直さなければ成らなかった作業が、『SUPER DEVICE MONITOR "T"』を使うと1バイト単位で編集が出来るので簡単に行える様に成ります。

256バイトを1セクターとしIPL-ROM、S-RAM、MIN-RAMなどが別々のディバイスとしてアクセス出来ます。
呼び出したセクターを、1バイト単位で変更・複写等の多彩なワープロ感覚の操作性を実現したエディット機能が使えます。
X68000標準フォーマット以外にも、OS/9やX1turboのフォーマット、外国製のMS-DOSのフォーマットなどの任意の(256〜4096bytes/セクター)フォーマットを選択して、アクセス出来ます。
S-RAMやIPLなど通常アクセス出来ない部分を含めてX68000内で呼び出せるメモリーは殆ど総てセクター単位でアクセス出来ます。
RS-232Cを使うと任意のボーレートでX68000同士は勿論、他機種にはその機種用の『SUPER DEVICE MONITOR "T"』を介して、特殊なデータ圧縮法により、データーによっては通常の32倍(理論値)の超高速で転送が行えます。2Dのデスク1枚分を1200ボーで1時間で転送出来ます。(X1のみ不可)
RS-232Cのボーレートの変更はボタン1つで簡単に出来ます。

SUPER DEVICE MONITOR "T"

X68000	5"	2HD	15,000円
X1	5"	2D	10,000円
X1turbo(2HDは受注生産)	5"	2D/2HD	13,000円
MZ-2500・MZ-2800	3.5"	2DD	13,000円

BLUE SKY Co.

▶お求めは全国の有名マイコンショップでどうぞ。
通信販売をご希望の方は当社へ直接、商品名・機種名・メディア名・住所氏名・電話番号を明記の上、現金書留にてお申し込みください。(送料無料)

株式会社 BLUE SKY
〒411 静岡県三島市加茂16-4
☎0559-72-6710

秋葉原価格でローンができます
電気の街秋葉原で
24年の信用!!

AVCフタバ

AUDIO VIDEO COMPUTER

☎03(253)7661

至新宿 / 電波ビル / 外堀通り / リビナ / AVCフタバ電機本社 / 日通ビル / 至神田 / 中央通り / 至上野 / 秋葉原駅 / ヤマギワ

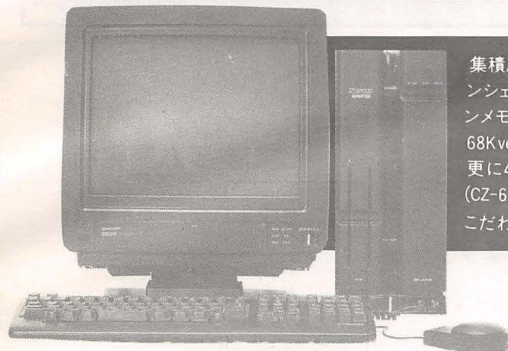
AVCフタバ電機
〒101 東京都千代田区外神田3-2-3
神田ユニオンビル ☎03-253-7661(代)

今すぐ もよりの電話から	仙 台 022-264-3704	名古屋 052-452-3271	広 島 082-295-6873
札 幌 011-611-5104	新 潟 0252-75-4175	大 阪 06-311-3931	福 岡 092-481-2494

X68000の情報のすべて!(当店はX68000の認定代理店です。お気軽にご相談下さい)

X68000 待望の新しい仲間登場!!

X68000 PERSONAL WORKSTATION
EXPERT・EXPERT HD

集積度を高めた"マンハッタンシェイプ"2Mバイトのメインメモリを標準実装、Human68K ver2.0搭載(CZ-602C)更に40MBのHDDを搭載(CZ-612C)あくまでもX68Kにこだわるマシン。

CZ-602C 標準価格¥356,000
CZ-612C 標準価格¥466,000
AVC特価

X68000 PERSONAL WORKSTATION
PRO・PRO HD

拡張I/Oスロットを4スロット標準装備、メインメモリ1MB、Human68K ver2.0搭載(CZ-652C)更に40MBのHDDを搭載(CZ-662C) 新しいX68Kの発見があるはずだ。
〔写真のモニタは別売です。〕

CZ-652C 標準価格¥298,000
CZ-662C 標準価格¥408,000
AVC特価

X68000 PERSONAL WORKSTATION
ACE・ACE HD

在庫有限

従来機も忘れずに!!

CZ-611C(HDDタイプ)
¥399,800
➡AVCフタバ特価
〔写真のモニタは別売です。〕

お勧めディスプレイコーナー 組合せは自由、価格はお気軽にご相談下さい。

CZ-612D 標準価格¥118,800 AVC特価
- 0.31mmドットピッチ
- TVチューナ搭載
- 3モードオートスキャン
- チルト台同梱

CZ-603D 標準価格¥84,800 AVC特価
- 0.31mmドットピッチ
- TVチューナ無し
- 3モードオートスキャン
- チルト台同梱

CZ-602D 標準価格¥99,800 AVC特価
- 0.39mmドットピッチ
- TVチューナ搭載
- 3モードオートスキャン
- チルト台同梱

CU-21CD 標準価格¥139,800 AVC特価
- 0.52mmドットピッチ
- TVチューナ無し
- 3モードオートスキャン
- チルト台取付不可

型番	品名	標準価格	販売価格
CU-14BD	ディスプレイ	¥ 64,800	AVCフタバ特価
CU-14ED	ディスプレイ	¥ 79,800	AVCフタバ特価
CU-14CD	ディスプレイ	¥ 84,800	AVCフタバ特価
CZ-860D	ディスプレイ	¥ 99,800	AVCフタバ特価
CZ-820D	ディスプレイ	¥ 79,800	AVCフタバ特価
DZ-880D	ディスプレイ	¥102,100	AVCフタバ特価
BF-68PRO	CRTフィルター	¥ 19,800	AVCフタバ特価
CZ-502F	FDD(2DD)	¥ 99,800	AVCフタバ特価
CZ-503F	FDD(2D)	¥ 49,800	AVCフタバ特価
CZ-6BE1A	1MB (増設RAMボード)	¥ 38,000	AVCフタバ特価
CZ-6BE2	2MB (増設RAMボード)	¥ 79,800	AVCフタバ特価
CZ-6BE4	4MB (増設RAMボード)	¥138,000	AVCフタバ特価
AN-160SP	アンプ内蔵スピーカー	¥ 59,800	AVCフタバ特価
CZ-8BS1	FM音源ボード	¥ 23,800	AVCフタバ特価
CZ-6BN1	スキャナ用パラレルボード	¥ 29,800	AVCフタバ特価
CZ-8PC2	熱転写プリンタ(24ドット)	¥ 69,800	AVCフタバ特価
CZ-8PC3	熱転写プリンタ(24ドット)	¥ 65,800	AVCフタバ特価
CZ-8PC4	熱転写プリンタ(48ドット)	¥ 99,800	AVCフタバ特価
AN-8TU	RGBシステムチューナ	¥ 33,100	AVCフタバ特価
CZ-8PK7	プリンタ(80桁)	¥122,000	AVCフタバ特価
CZ-8PK8	プリンタ(136桁)	¥152,000	AVCフタバ特価
CZ-8PK9	プリンタ(80桁)	¥ 89,800	AVCフタバ特価
CZ-6VT1	カラーイメージユニット	¥ 69,800	AVCフタバ特価
CZ-8BV2	カラーイメージボード	¥ 39,800	AVCフタバ特価
CZ-6BU1	ユニバーサルI/Oボード	¥ 39,800	AVCフタバ特価
CZ-6BG1	GP-IBボード	¥ 59,800	AVCフタバ特価
CZ-8TM1	モデム	¥ 29,800	AVCフタバ特価
CZ-8TM2	モデム	¥ 49,800	AVCフタバ特価
CZ-8NT1	トラックボール	¥ 13,800	AVCフタバ特価
CZ-6SD1	システムラック	¥ 44,800	AVCフタバ特価
CZ-6BF1	増設RS232Cボード	¥ 49,800	AVCフタバ特価
CZ-6BP1	数値プロセッサボード	¥ 79,800	AVCフタバ特価
CZ-6EB1	I/Oボックス	¥ 88,000	AVCフタバ特価
CZ-234LS	AI開発ツール	¥188,000	AVCフタバ特価
CZ-219SS	OS-9	¥ 29,800	AVCフタバ特価
CZ-227BS	TOP財務会計	¥200,000	AVCフタバ特価
CZ-213MS	MUSIC PRO-68K	¥ 18,800	AVCフタバ特価
CZ-214MS	SOUND PRO-68K	¥ 15,800	AVCフタバ特価
CZ-212BS	ビジネス PRO-68K	¥ 68,000	AVCフタバ特価
CZ-211LS	Cコンパイラ PRO-68K	¥ 39,800	AVCフタバ特価
CZ-141SF	NEW-Z BASIC	¥ 18,800	AVCフタバ特価
CZ-137SF	turboZ's STAFF	¥ 19,800	AVCフタバ特価
CZ-133SF	モデムターミナルソフト	¥ 25,800	AVCフタバ特価
	Z'STAFF PRO-68K	¥ 58,000	AVCフタバ特価
	kamikaze	¥ 68,000	AVCフタバ特価

X1G model30

X1Gの本格派セットFDD2基内蔵、専用カラーモニタはTVにも使用可能。

CZ-822C……¥118,000
CZ-820D……¥ 79,000
合計………¥197,000

特価 ¥79,800

お支払例 ¥ 7,603×12回 ¥ 5,228×18回
¥ 4,041×24回 ¥ 3,343×30回

X1turboZ III

X1ターボシリーズの独自の機能を全継承。VCCIゼロdB基準に適合させた。

CZ-888C……¥169,800
CZ-860D……¥ 99,800
合計………¥269,600

特価 ???

応談 価格はご相談に応じます。電話でお問い合せ下さい。

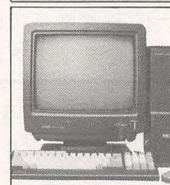
X1turboZ II

X1turboZの本格派セット。TV付2モードオートスキャンディスプレイ。

CZ-881C……¥179,800
CZ-880D……¥109,800
合計………¥289,600

特価 ???

応談 価格はご相談に応じます。電話でお問い合せ下さい。

X1twin

HEシステムを搭載、最上級ゲーム機とパソコンが合体。

CZ-830C……¥ 99,800
CZ-820C……¥ 79,800
合計………¥179,600

特価 ¥94,800

お支払例 ¥ 9,032×12回 ¥ 6,211×18回
¥ 4,800×24回 ¥ 3,363×36回

●頭金なし(手軽な電話クレジット)●製品先取り(お支払いは約1〜2ケ月後から)●低金利クレジット(1回の支払いは2,700円以上で3〜48回。ボーナス(併用も可)●カレッジクレジット(保証人なし。但し満20歳以上の学生の方)●18歳未満の方(ご両親が代理購入者としてお申し込み下さい)
●納期(通常の場合、当社に申込書が到着後1週間以内。特に人気のある商品や品薄の場合、少々納期が遅れることがありますので御了承下さい)
●完全保証(すべてメーカー保証書付。アフターケア万全)●全国代引(お届けした者に、代金をお支払いいただく方法です。但し手数料1,000円)

AM10時からPM7時まで受付 日曜・祝日も営業

●但し消費税(3%)は別途請求させていただきます。●分割回数は3回〜48回まで自由に選べます。

●セットの組合せは自由!広告に出ていない他の機種はお問合せ下さい。

パソコン・AV専門
O.A.ランド 大特価セール

OAランドで買わなきゃ損をする!

※4月1日より消費税を課税させていただきます。尚、表示価格は税別表示です。詳しくは、お電話下さい。

セール期間 ◀ '89 7・16 ➔ 8・16

NEW ランド特選 SHARP X68000EXPERT・EXPERT HDセット

X68000EXPERT HDセット

40MB HDD内蔵 2MB RAM

- CZ-612C……………………定価¥466,000
- CZ-612D……………………定価¥119,800
- MD-2HD 20枚サービス

ゲームソフト 5ゲームプレゼント

他店には負けません!! 合計定価¥585,800

現金大特価!! 安いぞ

X68000EXPERTセット

2MB RAM内蔵

- CZ-602C……………………定価¥356,000
- CZ-612D……………………定価¥119,800
- MD-2HD 20枚サービス

ゲームソフト 5ゲームプレゼント

OAランドで買わなきゃ損をする! 合計定価¥475,800

現金大特価!! 大推選!!

NEW X-1ターボZⅢセット

CRTクリーナー キーボードカバープレゼント

Ⓐセット

- CZ-888CBK…定価¥169,800
- CZ-880DBK… 定価¥109,800
- CZ-6ST1-B…定価¥ 5,800 (チルトスタンド)
- MD-2HD 20枚サービス

合計定価¥275,400

現金価格 特価中TEL下さい

安すぎて ゴメンなさい!

Ⓑセット

- CZ-888CBK…定価¥169,800
- CZ-830DBK…定価¥ 98,000
- CZ-6ST-1B…定価¥ 5,800 (チルトスタンド)
- MD-2HD 20枚サービス

合計価格¥273,600

合計価格 特価中TEL下さい

NEW SHARP X68000 PRO・PRO HDセット

ゲームソフト 5ゲームプレゼント

X68000PROセット

- CZ-652C………定価¥298,000
- CZ-612D………定価¥119,800
- MD-2HD 20枚サービス

合計定価¥417,800

現金特価!! TEL下さい。

X68000PRO-HDセット

- CZ-662C………定価¥408,000
- CZ-612D………定価¥119,800
- MD-2HD 20枚サービス

合計定価¥527,800

現金特価!! TEL下さい。

X68000 お買徳!! X-1TWIN

展示新同品

- CZ-611C(GY)
- CZ-611D(GY)

2セット限り

現金特価**¥32,800**

新同品

- CZ-830C 定価¥99,800

PCエンジン内蔵

現金特価**¥38,000**

周辺機器コーナー

X1用

- CZ-8BV2…定価¥ 39,800▶特価¥ 31,000
- CZ-8BR1…定価¥ 29,800▶特価¥ 23,000
- CZ-8DT2…定価¥ 44,800▶特価¥ 35,000
- CZ-8BS1…定価¥ 23,800▶TEL下さい
- CZ-8TM2…定価¥ 49,800▶特価¥ 38,000
- CZ-8EB3…定価¥ 33,800▶特価¥ 27,000

X68000用

- CZ-6PU1A‥定価¥ 38,000▶特価¥ 30,000
- CZ-6BM1…定価¥ 26,800▶特価¥ 21,000
- CZ-6BE1…定価¥ 88,000▶特価¥ 69,800
- CZ-6VT1…定価¥ 69,800▶TEL下さい
- CZ-8NS1…定価¥188,000▶特価¥149,000
- CZ-6BC1…定価¥ 79,800▶特価¥ 63,000

プリンターセットコーナー

1. CZ-6PU1(カラービデオプリンター)定価¥198,000▶特価¥152,000
2. CZ-8PC3(カラープリンター)……定価¥ 65,800▶特価¥ 53,000
3. CZ-8PK8(ドットプリンター)………定価¥152,000▶特価¥115,000
4. CZ-8PK7(ドットプリンター)………定価¥122,000▶特価¥ 93,000
5. PC-PR201TH(カラープリンター)・定価¥145,000▶特価¥103,000
6. PC-PR201G(ドットプリンター)…定価¥158,000▶特価¥ 99,000

その他、周返機器・プリンター ソフトウェアー 20%〜25% OFF!!

X68000用ソフトウェアー・コーナー

1. CZ-212BS(BUSINESS)………定価¥ 68,000▶特価¥ 53,000
2. CZ-220BS(DATA)……………定価¥ 58,000▶特価¥ 45,000
3. CZ-215MS(Sampling)…………定価¥ 17,800▶特価¥ 13,800
4. CZ-221HS(NEW Print Shop)‥定価¥ 10,800▶特価¥ 15.500
5. CZ-227BS(TOP財務会計)……定価¥200,000▶特価¥158,000
6. CZ-226BS(CARD)……………定価¥229,800▶特価¥ 23,000
7. CZ-223CS(Communication)…定価¥ 19,800▶特価¥115,500
8. CZ-213MS(MUSIC)……………定価¥ 18,800▶特価¥ 14,800
9. CZ-211LS(C compiler)………定価¥ 39,800▶特価¥ 31,000
10. C-TRACE(キャスト)……………定価¥ 68,000▶特価¥ 52,000
11. EW(イースト)…………………定価¥ 38,000▶特価¥ 29,000

■ハードディスク ■特価品もありますのでTEL下さい。

- アイテック IT-MJ4(I/F付)…………特価¥98,000
- アイテック IT-MJ4 C(I/F付)……特価¥109,000
- ウィンテック HD-404HS(I/F付)…特価¥108,000
- コンピュータ CRC-MH4(I/F付)……特価¥70,000
- スナイパー SR-340Ⅱ(I/F付)………特価¥78,000
- アイテック ITH-320S(I/F付)………特価¥79,800
- ウィンテック HD-202(I/F付)………特価¥58,000
- スナイパー SR-520(I/F付)…………特価¥55,000
- コンピュータ CRC-HD2A(I/F付)……特価¥62,000
- ロジテック LHD-32NR(I/F付)………特価¥80,000

今月の特価品 各一台限り その他、いろいろありますのでTEL下さい!!

■A紙品(美品・POP品) ■B級品(キズ少々) ■C級品(キズ有り)

		A級品	B級品	C級品
X68000シリーズ	CZ-611C	¥250,000より	¥245,000	¥238,000
	CZ-652C	¥219,000より	¥212,000	¥203,000
	CZ-611D	¥ 90,000	¥ 86,000	¥ 80,000
	CZ-603	¥ 58,000	¥ 55,000	¥ ——
X-1シリーズ	CZ-888C	¥ 99,800より	¥ 90,000	——
	CZ-822C	¥ 24,000より	¥ 20,000	——
	CZ-880D	¥ 75,000	¥ 71,000	——
	CZ-830C	¥ 37,000	¥ 33,000	——
X-1プリンター	CZ-8PC3	¥ 48,000	¥ 45,000	¥ 42,000
	CZ-7PK7	¥ 83,000	——	——
	CZ-8PK8	¥109,000	¥105,000	——
	CZ-6PV1	¥138,000	¥134,000	¥125,000

その他、いろいろありますので、TELください。

中古パソコン(価格・在庫は変動します。予約は5日以内といたします。)

- PC-9801VX2ト…………¥220,000より
- PC-9801VX2……………¥195,000より
- PC-9801VM2……………¥158,000より
- PC-9801VF2……………¥ 98,000より
- PC-9801M2………………¥138,000より
- PC-9801F2………………¥ 78,000より
- PC-9801UV21……………¥138,000より
- PC-98LTM1(640KB)‥¥ 89,000より
- PC-286モデル0……‥¥168,000より
- PC-286V-STD…………¥202,000より
- X-68000…………………¥188,000より
- PC-8801mkⅡ30……¥ 35,000より
- PC-8801mkⅡSR……¥ 73,000より
- PC-8801mkⅡFR30‥¥ 68,000より
- PC-8801mkⅡMR……¥ 88,000より
- PC-88VA……………¥148,000より
- PC-8801mkⅡFH30‥¥ 85,000より
- PC-8801FA……………¥108,000より
- X-1Gモデル30………¥ 25,000より
- X-1ターボⅡ…………¥ 68,000より
- FM-77D2………………¥ 28,000より
- FM-77AV2………………¥ 42,000より
- FM-77AV20……………¥ 52,000より

通信販売のご案内

—— 全国通販 ——

■銀行振込で申し込みの方は商品名及びお客様の住所・氏名・電話番号をお知らせ下さい。

〔振込先〕第一勧業銀行 渋谷支店
普通No.1163457 ㈱オーエーランド

■現金書留で送金されるお客様は電話番号と商品名、数量を明記して同封して下さい。■クレジットでご購入を希望される方は申し込み用紙をお送り致しますのでご記入の上返送して下さい。20才以上の方は、原則として保証人不要です。クレジットは1〜60回払で月々5,000円よりご自由に設定できます。

品川、目黒方面 首都高速3号線 渋谷駅 井の頭線渋谷駅 道玄坂 O.A.ランド ヤマハ 文 109 J&P 神泉駅 明治通り 山手線 西武百貨店 東急百貨店

- 下取・買取は電話で見積りしております。責任を持って下取りさせて頂きます。
- ご注文、お問合せは…毎日午前10時から午後7時まで
- 商品のお届けは…入金確認後、即日発送致します。

㈱オーエーランド

〒150東京都渋谷区円山町20-4 第5日新ビル1F

☎(03)770-8855

FAX(03)770-7080

関東エリアの送料は、1個につき¥1,000です。

クレジット
金利大幅
ダウン!!

COMPUTER BANK

J-DMA 安心と信頼のシステムで新時代を切り開く

話題の新製品が全国どこでも電話で買える!!
03(797)1221

X68000

EXPARTシリーズ
・PROシリーズ新登場!!

・オリジナルOS「Human68k ver. 2.0」を搭載
・40MBハードディスクドライブを内蔵
・メインメモリ2MB標準装備(EXPERTシリーズ)
・拡張I/Oスロット4スロット内蔵(PROシリーズ)

☆注文No.A-0821
SHARP CZ-602C ¥356,000
SHARP CZ-602D ¥ 99,800
標準価格合計 ¥455,800
現金特別価格 ~~¥455,800~~
大特価にて提供中

☆注文No.A-0822
SHARP CZ-612C ¥466,000
SHARP CZ-602D ¥ 99,800
標準価格合計 ¥565,800
現金特別価格 ~~¥565,800~~
大特価にて提供中

☆注文No.A-0823
SHARP CZ-652C ¥298,000
SHARP CZ-602D ¥ 99,800
標準価格合計 ¥396,800
現金特別価格 ~~¥396,800~~
大特価にて提供中

☆注文No.A-0824
SHARP CZ-662C ¥408,000
SHARP CZ-602D ¥ 99,800
標準価格合計 ¥507,800
現金特別価格 ~~¥507,800~~
大特価にて提供中

当社はX68000 PRO SHOPです。

■周辺機器　大特価にて提供中

品番	品名・内容	定価
CZ-602D	15型カラーディスプレイテレビ	¥ 99,800
CZ-612D	15型カラーディスプレイテレビ	¥119,800
CZ-603D	14型カラーディスプレイ	¥ 84,800
CZ-6ST1	601D・611D用チルトスタンド	¥ 5,800
CU-21CD	21型カラーディスプレイ	¥139,800
CZ-6TU	RGBシステムチューナー	¥ 33,100
BF-68PRO	601・611・603用CRTフィルター	¥ 19,800
CZ-6VT1	カラーイメージユニット	¥ 69,800

型番	品名・内容	定価
CZ-8NS1	カラーイメージスキャナ	¥188,000
CZ-6BN1	スキャナ用パラレルボード	¥ 29,800
CZ-6BE1A	1MB増設RAMボード(内蔵用)	¥ 38,000
CZ-6BE2	2MB増設RAMボード(内蔵用)	¥ 79,800
CZ-6BE4	4MB増設RAMボード(内蔵用)	¥138,000
CZ-6BU1	ユバーサルI/Oボード	¥ 39,800
CZ-6BG1	GP-IBボード	¥ 59,800
CZ-6BF1	増設用RS-232Cボード(2ch)	¥ 49,800

型番	品名・内容	定価
CZ-6BP1	数値演算プロセッサボード	¥ 79,800
CZ-6BC1	FAXボード	¥ 79,800
CZ-6BM1	MIDIボード	¥ 26,800
CZ-6EB1	拡張I/Oボックス(4スロット)	¥ 88,000
CZ-6PV1	カラービデオプリンタ	¥198,000
CZ-6BU1	ユバーサルI/Oボード	¥ 39,800
CZ-620H	ハードディスクユニット(20MB)	¥178,000
AN-S100	アンプ内蔵スピーカーシステム(2本1組)	¥ 36,800

■ソフトウェア　大特価にて提供中

メーカー名	型番	品名・内容	定価
SHARP	CZ-212BS	BUSINESS PRO-68K	¥68,000
SHARP	CZ-220BS	DATA PRO-68K	¥58,000
SHARP	CZ-226BS	CARD PRO-68K	¥29,800
SHARP	CZ-214MS	SOUND PRO-68K	¥15,800
SHARP	CZ-213MS	MUSIC PRO-68K	¥18,800
SHARP	CZ-215MS	Sampling PRO-68K	¥17,800

メーカー名	型番	品名・内容	定価
SHARP	CZ-237MS	Musicstudio PRO-68K	¥25,800
SHARP	CZ-247MS	MUSIC PRO-68K〔MIDI〕	¥28,800
SHARP	CZ-221HS	NEW Print Shop PRO-68K	¥19,800
SHARP	CZ-223CS	Communication PRO-68K	¥19,800
SHARP	CZ-211LS	C compiler PRO-68K	¥39,800
SHARP	CZ-219SS	OS-9/6800	¥29,800

メーカー名	型名	品名・内容	定価
イースト	EW	日本語ワープロ	¥38,000
アンスコンサルタンツ	彩CRONE68K	グラフィックツール	¥58,000
CAST	C-TRACE68	グラフィックツール	¥68,000
ツァイト	Z'sSTAFF PRO	グラフィックツール	¥58,000
電波新聞社		ドラゴンスピリット	¥ 8,800
テクノソフト		サンダーフォースII	¥ 9,800

●どこよりもお得な高額下取り実施中!!　セットの組合わせは自由自在、ぜひご相談下さい。

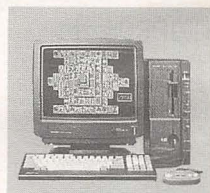
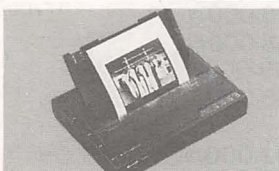
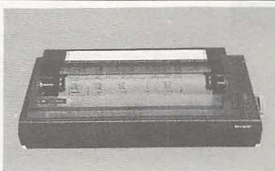
X1 turbo Z III

画像取り込み、ビデオ編集、ステレオFM
音源、多才な機能でひろがるアートワーク。

☆注文No.A-0825
SHARP CZ-888C-BK ¥169,800
SHARP CZ-860D-BK ¥ 92,200
標準価格合計 ¥262,000
現金特別価格 ~~¥262,000~~
大特価にて提供中

X1 twin

HEシステム(PC Engine)
搭載で楽しさ2倍

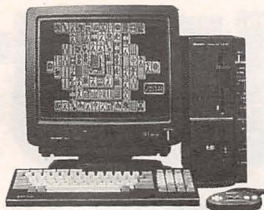
☆注文No.A-0826
SHARP CZ-830C-BK ¥ 99,800
SHARP CZ-830D-BK ¥ 90,600
標準価格合計 ¥190,400
現金特別価格 ~~¥190,400~~
大特価にて提供中

●どこよりもお得な高額下取り実施中!!　セットの組合わせは自由自在、ぜひご相談下さい。

☆注文No.B-0823
SHARP CZ-8PC3 ¥65,800
現金特別価格 ~~¥65,800~~
大特価にて提供中
■お支払例
①¥10,000×6回〔ボーナス〕無し
②¥ 3,200×20回〔ボーナス〕無し

☆注文No.B-0824
SHARP CZ-8PK6 ¥159,000
現金特別価格 ¥59,800
■お支払例
①¥6,500×10回〔ボーナス〕無し
②¥3,400×24回〔ボーナス〕無し

☆注文No.B-0825
SHARP CZ-8PC4 ¥99,800
現金特別価格 ~~¥99,800~~
大特価にて提供中
■お支払例
①¥9,500×10回〔ボーナス〕無し
②¥3,000×36回〔ボーナス〕無し

☆注文No.B-0832
SHARP AN-8TU ¥33,100
現金特別価格 ~~¥33,100~~
大特価にて提供中

全商品保証付 中古も6ヶ月の保証期間だから安心です。
クレジットでOK カレッジクレジットも取扱います。
全国無料配送 お買上1万円以上、配達料はいただきません。
日曜配達可 留守の多い方でも安心です。
ショールーム Xシリーズ展示中。
高額買取り 電話1本で即、現金お支払い。
代金引換えシステム 商品到着時の代金支払いでOK。
ボーナス一括払い 商品は即お手元へ、お支払いはボーナス時に。

中古在庫リスト

超優良中古パソコンが電話一本で買える!!
03(797)1221

SHARP
CZ-830C
(X-1Twin)
¥99,800➡**¥46,000**

SHARP
CZ-820C
(X-1Gモデル10)
¥69,800➡**¥9,000**

SHARP
CZ-611CGY
(X68000ACEHD) 新品同様
¥399,800➡**¥278,000**

X68000 ACEHD
ディスプレセット
(本体+CZ-611DGY) 新品同様
¥533,800➡**¥370,800**

SHARP
CZ-8PK6 新品同様
(15インチ漢字プリンタ)
¥159,000➡**¥59,800**

SHARP
CU-14GD 新品同様
(PC用14インチ400Lアナログディスプレイ)
¥69,800➡**¥54,800**

SHARP
CZ-822C
(X-1Gモデル30本体) 新品同様
¥118,000➡**¥29,800**
X-1Gモデル30RFコンバータセット
(本体+AN-58C) 新品同様
¥120,980➡**¥32,600**

SHARP

本体

商品	定価	価格
CZ-812C(X-1F model20)	¥139,800	¥26,000
CZ-822C(X-1F model30)	¥118,000	¥28,000
CZ-822CB(X-1G model30) 新品同様	¥118,000	¥29,800
CZ-830C(X-1 Twin)	¥99,800	¥46,000
CZ-601C(X68000ACE) 新品同様	¥319,800	¥238,000
CZ-611C(X68000ACEHD) 新品同様	¥399,800	¥278,000
MZ-1500	¥89,800	¥15,000

ディスプレイ

商品	定価	価格
12M-18B(12″グリーン4050文字)	¥44,800	¥20,000
14M-142C(14″カラー2000文字)	¥99,800	¥22,000
CU-14BD(14″カラー2000/4000文字)	¥64,800	¥42,000
CU-14GD(14″カラー4050文字) 新品同様	¥69,800	¥54,800
CU-14FD(14″カラー4050文字) 新品同様	¥74,800	¥59,800
CZ-880DBK(14″カラー2000/4000文字TV) 新品同様	¥102,000	¥75,000
CZ-603D(15″カラーX68000) 新品	¥84,800	¥68,000
CZ-611D(15″カラー3モードスキャンディスプレイTV) 新品同様	¥134,000	¥92,800

ディスクドライブ・プリンタ・他

商品	定価	価格
CZ-52F(X1用内臓ドライブ)	¥34,800	¥9,800
CZ-503F(X1用増設1ドライブ)	¥49,800	¥25,000
CZ-81P(CZ-801C専用カラープロッタプリンタ)	¥54,800	¥10,000
CZ-8PK4(10″24ドット漢字プリンタ)	¥158,000	¥45,000
CZ-8PK6(15″24ドット漢字プリンタ) 新品	¥159,000	¥59,800
CZ-8PK7(10″24ドット漢字プリンタ)	¥122,000	¥52,000
MZ-1P07(10″ドットプリンタ)	¥79,800	¥22,000
MZ-1P17(10″24ドットカラー漢字サーマルプリンタ)	¥79,800	¥30,000
CZ-8SS2(システムスタンド) 新品	¥5,500	¥3,500
CZ-8BS1(FM音源ボード) 新品	¥23,800	¥20,000

その他各種在庫をとりそろえております。御気軽にお問い合わせ下さい。

6つの安心のアフターサービス

1 C.B.クラブ

■あなたも今すぐ会員に!!

当社で商品をお買い上げの方全員に、**C.B.クラブカードを無料**でお送り致します。このカードをお持ちの方なら次の買い換え時や、付属品の購入時に会員特別価格でご購入になれます。

2 C.B.サポートホットライン

☎03(797)1234

■トラブルへの対応!!

当社でコンピュータをお買い上げいただいたお客様に万一、トラブルが発生した場合、このホットラインで親切に対応いたします。

3 C.B.レスキューシステム

■迅速なサポート体制!!

お客様のお手元でトラブルが発生した場合、当社より引取りにお伺い致します。万一、お買いになった機械が故障しても安心です。

4 C.B.クイック・チェンジシステム

■新品交換体制も万全!!

お買い上げになったパソコンが、万一初期不良でも安心です。商品到着後7日以内にご連絡いただければ、新品と交換致します。

5 RX2アフターサポート

■PC-9801愛好家にお得です!!

NEC RX2をお買い上げいただいたお客様に保証期間中、万一故障があった場合無料で代品を貸出します。

6 C.B.Q&Aホットライン

☎03(797)1233

■素朴な疑問何でもどうぞ!!

ハードウェア、ソフトウェアに関するご質問なら内容を問わずどなたからでも親切に、ご相談をお受け致しております。

●コンピュータを売りたい方、査定をご希望の方、その他買取りに関するご相談は●

買取り専用デスク 03(797)1231

●電話一本で高額下取り、即商品はお手元へ!
●あなたの不要になったパソコンを電話一本で査定し買取ります。
●掲載の商品以外も取り扱っております。
●ビジネスソフトスクール受講者受付中!お気軽にお電話下さい。

▼本社注文デスク
03(797)1221

コンピュータバンク

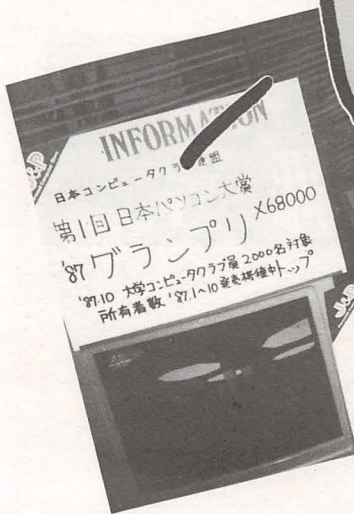
株式会社パシフィックコンピュータバンク 〒150 東京都渋谷区渋谷1-6-8 井上ビル 営業時間/平日AM9:30～PM9:00 土・休日AM9:30～PM8:00 年中無休

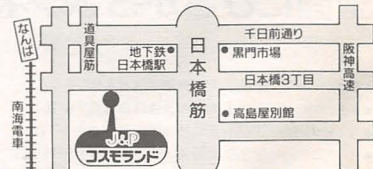
●クレジット価格に消費税は含まれておりますが、現金特別価格には含まれておりません。別途消費税がかかります。

コンピュータのことなら…

システムで提案する

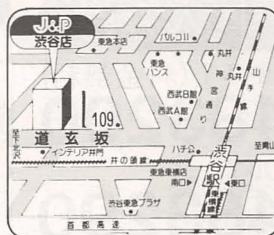
パソコンとOA機器の大型専門店J&Pチェーンは、
たんにハードの販売だけではなく、
使い方や目的に応じてシステムでご提案いたします。
さらにセミナー、イベント、パソコン通信ネットワークなど、
パソコンの世界を限りなく拡げます。

緊急速報！
第2回
オールジャパン
オリジナルソフト
博覧会開催！
入場無料
日時/7月29日(土)・30日(日)
(29日(土) AM10:00〜PM7:00)
(30日(日) AM10:00〜PM6:00)
全国のマニアの夢が集まった！
あんな理想、こんな可能性、メーカーには絶対
マネのできない学生ならではの自由な発想で
ボクらの夢のソフトが大集合！
これを見なけりゃ、マニアじゃない！
INFORMATION
第1回日本パソコン大賞
'87グランプリ X68000
INFORMATION
大阪大学
X-68000
CGAデモ
⊙京都大学・大阪大学・神戸大学・和歌山大学・京都教育大学・
関西大学などの全国有名大学コンピュータクラブ、
Final Tear Zなどの全国有名一般
コンピュータクラブなど、選抜された
全国の実力クラブのオリジナルソフト
の展示、デモ説明、無料コピーサービス実施。
⊙全国有名同人ソフトの即売会。
⊙来場者の投票による。
1989年度、全国No.1の
コンピュータクラブ決定。
●主催/日本コンピュータクラブ連盟
●協賛/日コン連企画株式会社
●後援/上新電機株式会社
Joshin Computer Store
J&P
コスモランド
ニューメディア時代の暮屋／／J&P
パソコンとビジュアルの／／J&Pコスモランド
Joshin Computer Store
J&P
大阪難波 コスモランド
大阪市浪速区難波中2丁目1番17号(〒556)
☎(06) 634-3111
なんば
道具屋筋
地下鉄日本橋駅
日本橋筋
千日前通り
黒門市場
日本橋3丁目
高島屋別館
阪神高速
南海電車
J&P コスモランド
なんば駅より3分、
地下鉄日本橋駅下車すぐ

パソコン通信

J&P

J&P HOT LINE でもお申し込みいただけます。

メールショッピングのお申し込みは J&P 渋谷店で承ります。

全国無料配達

メールショッピング

東京都渋谷区道玄坂2丁目28番4号(〒150)
☎(03)496-4141〈水曜定休〉

●電子手帳

X8-1

■シャープ電子手帳PA-8500
メーカー標準価格¥28,000 **¥24,800**
約660人分のデータ管理にカレンダー・スケジュール・メモ・計算・時計機能搭載。さらにオプションカードで面白さが広がります。

〈PA-8500を面白くするICカード〉

X8-2
- シャープPA-7C40
英和辞書カード 4行表示 **¥16,000**
本格派の辞書に匹敵する4万語を収録。
- シャープPA-7C41
国語辞典カード 4行表示 **¥16,000**
読みを入力すれば漢字の変換、意味、熟語、用語、類語、対語を表示。
- シャープPA-7C43
珠玉格言集カード 4行表示
メーカー標準価格¥10,000
¥9,000
約1700句の名言・格言を収録。中国古典、日本古典より精選。
- シャープPA-7C10
電話帳/住所録カード 4行表示
メーカー標準価格¥10,000
¥9,000
電話帳なら約330人、住所録なら約170人分を記憶。メモリー・バックアップ付。
- シャープPA-7C3 メーカー標準価格¥7,000
電訳機、6ケ国語カード **¥6,300**
日本語、中国語、韓国語、英語、フランス語、スペイン語の会話文約350例と単語約610語を収録

X8-3

■ハル研究所 HAL-CATCH
¥11,000 パソコンで電子手帳のデータ管理を実現。大きな画面で整理したデータを再度電子手帳へ転送可能。
●電子手帳は別売

●パソコン通信

J&P HOTLINE入会キットがセットなので買ったその日からアクセス可能

X8-4

■オムロンモデム MD-12FS
＋ J&P HOTLINEスタータキット
¥22,800
AC・DC両用タイプなのでラップトップパソコンに最適。1200・360bps全二重。

X8-5

高速通信セット

■アイワモデム
PV-A24MNP5
＋ J&P HOTLINEスタータキット
¥49,800
2400・1200・300bps全二重。高速通信に対応した人気モデムにJ&P HOTLINE入会キットをセット。

●ハンディーコピー

X8-6

■ゼロックスハンディコピー
写楽アルファII
¥56,800
104mm幅、拡大縮小機能のついた実務機。

●パソコンアクセサリー

X8-7

■5インチ2HD
ディスケット
スリーエム MD2D256EX
30枚で **¥4,200**

Datalife MF-2HD256 PRO
X8-8

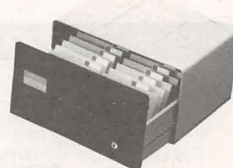
■3.5インチ2HD
ディスケット
データライフ
MF-2HDPRO
20枚で **¥8,500**

X8-10

■フロッピィケース
5インチ用
エレコム FP5-180
メーカー標準価格¥5,800
¥5,000

X8-9

■OAライト
MEIKO EP-303
メーカー標準価格12,800
¥6,980
パソコンライトU
遮光板付

JOYCONT TURBO-3 MACROTEC

■ジョイカード X8-12
スピタル JOYCONT-turbo-3
メーカー標準価格¥2,200
¥2,000 X-6800用 ダブルトリガーで即連射！

X8-11

■フロッピィケース3.5インチ用
エレコム FP35-180 メーカー標準価格¥5,800
¥5,000

■キーボード
防塵カバー X8-13
エレコムX-6800エキスパート
プロ用があります。
¥2,800
シリコン製。PKB-X68

お申し込み方法

右の注文書にご希望商品の注文No.および必要事項ご記入の上、現金書留にて J&P 渋谷店までお申し込みください。現金受領後、発送いたします。
また、J&P HOTLINE会員の方は、ショッピングコーナーでもお申し込みいただけます。

●記載商品以外のご注文も承ります。
詳しくはお電話にてお問い合わせ下さい。
☎(03)496—4141 定休：毎週水曜日

キリトリ線

現金書留申込み用紙

おところ 〒□□□-□□ TEL ( )	注文No.	数量	金額
	X8- ( )		円
	X8- ( )		円
	合計		円
おなまえ 様	お手持ちのパソコン		

お申込み先：東京都渋谷区道玄坂2丁目28番4号(〒150) J&P 渋谷店メールショッピング係

安心と信頼の誌上ショッピング メディアショップ

お申込みは今すぐ電話かハガキで!!

株式会社 メディアショップ ハイランド　〒239 神奈川県横須賀市ハイランド3-9-6

電話でのお申込みは

お申し込みはフリーダイヤルで(料金無料)

0120-483290

お問合せは専用ダイヤルで

☎0468-483290

年中無休AM10時～PM10時

ハガキでのお申込みは

〒239
神奈川県横須賀市ハイランド3-9-6
(株)メディアショップハイランド
Oh!X 係

申込書
- 商品名(商品番号)
- 支払回数
- お名前
- 生年月日
- ご住所、電話番号
- お勤め先
  名称、住所、電話番号

通信販売のお申込み方法

▶現金一括でお申込みの方
- 商品名(商品番号)及び、住所、氏名、電話番号、ご覧の雑誌名をご記入の上、代金を現金書留でお送り下さい。
- 振込をご希望の方は、必ずお振込前にお電話又はおハガキで、お知らせ下さい。
  〈銀行振込〉協和銀行・久里浜支店　当座No.2945
  〈郵便振替〉横浜9-42177

▶クレジットでお申込みの方
- 電話かハガキでお申込み下さい。
  クレジット申し込み用紙をお送り致しますので、ご記入の上、当社へお送り下さい。

SHARP X68000 EXPERT

- CZ-602C(FDタイプ) 標準価格 356,000円
- CZ-612C(HDタイプ) 標準価格 466,000円
- CZ-602D(ディスプレイテレビ) 標準価格 99,800円
- CZ-612D(ディスプレイテレビ) 標準価格 119,800円
- CZ-603D(ディスプレイ) 標準価格 84,800円

SHARP X68000 PRO

- CZ-652C(FDタイプ) 標準価格 298,000円
- CZ-662C(HDタイプ) 標準価格 408,000円
- CZ-602D(ディスプレイテレビ) 標準価格 99,800円
- CZ-612D(ディスプレイテレビ) 標準価格 119,800円
- CZ-603D(ディスプレイ) 標準価格 84,800円

X68000 オリジナルグッズ プレゼント!!

- X68000 スポーツタオル
- X68000 ビジネスバッグ
- X68000 ポーチ
- X68000 マウスパット

御買上げのお客様に、X68000オリジナルグッズを1点もれなくプレゼント。

EXPERT グラフィックス

- CZ-612C(本体)……466,000円
- CZ-612D(ディスプレイテレビ)……119,800円
- CZ-8NS1(イメージスキャナー)……188,000円
- CZ-6BN1(パラレルボード)……29,800円
- AP-800(48ドットカラープリンタ)……97,800円
- #8226(インターフェイスケーブル)……8,800円
- A-400HP(ビデオデッキ)……104,800円
- CZ-221HS(NEW Print SHOP)……19,800円
- C-TRACE68(レイトレーシングソフト)……68,000円

標準価格1,102,800円

商品番号 227	一括払価格 特別価格
初回16,500円・12,600円×47回	ボーナス60,000円×8回
初回14,500円・10,600円×59回	ボーナス50,000円×10回

EXPERT サウンド〔MIDI〕

- CZ-602C(本体)……356,000円
- CZ-602D(ディスプレイテレビ)……99,800円
- AN-160SP(アンプ内蔵スピーカーシステム)55,300円
- CZ-6BM1(MIDIボード)……26,800円
- MT-32(MIDI音源モジュール)……69,000円
- D-10(キーボード)……128,000円
- CZ-215MS(Sampling PRO68K)……17,800円
- CZ-247MS(MUSICPRO68K MIDI)……28,800円

標準価格 781,500円

商品番号 228	一括払価格 特別価格
初回11,000円・10,100円×47回	ボーナス50,000円×8回
初回14,100円・8,700円×59回	ボーナス40,000円×10回

EXPERT 通信・パソコンFAX

- CZ-612C(本体)……466,000円
- CZ-603D(ディスプレイ)……84,800円
- CZ-8TM2(モデムユニット)……49,800円
- VP-2000(136桁カラー漢字ドットプリンタ)……156,000円
- #8226(インターフェイスケーブル)……8,800円
- CZ-6BC1(FAXボード)……79,800円
- CZ-223CS(Communication)……19,800円

標準価格 865,000円

商品番号 219	一括払価格 特別価格
初回12,000円・9,700円×47回	ボーナス50,000円×8回
初回13,200円・8,400円×59回	ボーナス40,000円×10回

PRO ワープロ

- CZ-652C(本体)……298,000円
- CZ-603D(ディスプレイ)……84,800円
- VP-2000(136桁カラー漢字ドットプリンタ)……156,000円
- #8226(インターフェイスケーブル)……8,800円
- EW(日本語ワープロソフト)……38,000円

標準価格 585,600円

商品番号 221	一括払価格 特別価格
初回10,700円・7,300円×47回	ボーナス30,000円×8回
初回8,800円・7,000円×59回	ボーナス20,000円×10回

PRO データベース

- CZ-662C(本体)……408,000円
- CZ-612D(ディスプレイテレビ)……119,800円
- CZ-6VT1(カラーイメージユニット)……69,800円
- VP-900(80桁カラー漢字ドットプリンタ)……126,000円
- #8226(インターフェイスケーブル)……8,800円
- CZ-220BS(DATA PRO68K)……58,000円
- CZ-226BS(CARD PRO68K)……29,800円

標準価格820,200円

商品番号 229	一括払価格 特別価格
初回12,900円・8,900円×47回	ボーナス50,000円×8回
初回10,200円・7,800円×59回	ボーナス40,000円×10回

X1, X1turbo, X68000シリーズ用周辺機器

カラービデオプリンタ

- CZ-6PV1
  パソコンやビデオ機器に対応。64階調(485×480ドット)で再現する、昇華性染料熱転写方式を採用。

標準価格 198,000円

商品番号 149	一括払価格 特別価格
24回	初回9,600円・7,500円×23回
36回	初回5,500円・5,200円×35回

カラー イメージ スキャナー

- CZ-8NS1
  高速、高精度でハイレベルな画像入力を実現 最大A4サイズの原稿をフルカラー読み取り可能

標準価格 188,000円

商品番号 188	一括払価格 特別価格
24回	初回8,100円・7,200円×23回
36回	初回7,400円・4,900円×35回

48ドット熱転写カラー漢字プリンタ

- CZ-8PC4
  精彩で瞭字のない高品位印字。美文書もアートワークも鮮やかに、美しさの48ドットカラープリンタ。

標準価格 99,800円

商品番号 216	一括払価格 特別価格
12回	初回7,600円・7,400円×11回
24回	初回4,200円・3,900円×23回

21型カラーディスプレイ

- CU-21CD
  応用分野を広げるワイド画面。3モードマルチスキャン採用アナログカラーディスプレイ。

標準価格 139,800円

商品番号 217	一括払価格 特別価格
24回	初回7,300円・5,300円×23回
36回	初回7,000円・3,600円×35回

商品名	型式	標準価格	販売価格
14型カラーディスプレイ	CZ-603D	84,800	71,900
RGBシステムチューナー	CZ-6TU	33,100	29,300
CRTフィルター	BF-68PRO	19,800	16,300
熱転写カラープリンタ	CZ-8PC3	65,800	54,000
漢字プリンタ(80桁)	CZ-8PK9	89,800	72,700
漢字プリンタ(80桁)	CZ-8PK7	122,000	98,400
漢字プリンタ(136桁)	CZ-8PK8	152,000	122,800
ハードディスク(20MB)	CZ-620H	178,000	143,700
増設用HDD(40MB)	CZ-64H	120,000	100,200
モデムユニット	CZ-8TM2	49,800	42,100

商品名	型式	標準価格	販売価格
カラーイメージユニット	CZ-6VT1	69,800	59,000
スキャナ用パラレルボード	CZ-6BN1	29,800	25,200
1MB増設RAM	CZ-6BE1	35,000	29,500
1MB増設RAM	CZ-6BE1A	38,000	32,100
2MB増設RAM	CZ-6BE2	79,800	67,300
4MB増設RAM	CZ-6BE4	138,000	116,400
ユニバーサルI/Oボード	CZ-6BU1	39,800	33,600
GP-IBボード	CZ-6BG1	59,800	50,400
増設用RS-232Cボード	CZ-6BF1	49,800	42,000
数値演算ボード	CZ-6BP1	79,800	67,300

商品名	型式	標準価格	販売価格
FAXボード	CZ-6BC1	79,800	67,300
MIDIボード	CZ-6BM1	26,800	23,200
拡張I/Oボックス	CZ-6EB1	88,000	74,200
システムラック	CZ-6SD1	44,800	37,800
スピーカーシステム	AN-160SP	55,300	48,500
カラーイメージボードII	CZ-8BV2	39,800	33,500
立体映像セット	CZ-8BR1	29,800	24,600
インテリジェントコントローラ	CZ-8NJ2	23,800	20,100
FM音源ボード	CZ-8BS1	23,800	20,100
フロッピーディスクユニット	CZ-503F	49,800	39,200

商品名	型式	標準価格	販売価格
DATA PRO68K	CZ-220BS	58,000	49,300
CARD PRO68K	CZ-226BS	29,800	25,400
Sampling PRO68K	CZ-215MS	17,800	15,300
NEW Print SHOP	CZ-221HS	19,800	16,400
Communication	CZ-223CS	19,800	16,900
C compiler	CZ-211LS	39,800	34,500
Musicstudio	CZ-237MS	25,800	22,200
MUSIC(MIDI)	CZ-247MS	28,800	24,600
OS-9/X68000	CZ-219SS	29,800	25,400
WORD PRO-68K	CZ-225BS		

今月の特選お買得品(限定)

SHARP X68000 ACE-HD

- CZ-611C
  X68000にHDモデル登場。ますます熱くなる、パーソナルワークステーション。
- CZ-611D
  15型カラーディスプレイテレビ。

標準価格 544,800円

商品番号 183	一括払価格 特別価格
48回	初回11,500円・10,500円×47回
60回	初回9,600円・8,800円×59回

SHARP X1 turbo Z III

- CZ-888C
  画像取り込み、ビデオ編集ステレオFM音源、多彩な機能で広がるアートワーク。ADVANNCED TURBO
- CZ-860D
  14型カラーディスプレイテレビ。

標準価格 262,000円

商品番号 200	一括払価格 特別価格
24回	初回11,600円・9,600円×23回
36回	初回8,400円・6,600円×35回

安心と信頼 メディアショップ ハイランド

①完全保証 全国どこでもアフターケアOK
②全国無料配送 日曜配送可能
③支払回数は 予算に応じ3～36回 ボーナス併用可
④消費税 広告は全て消費税込みの価格で表示してあります
⑤FAXでも注文OK FAX:0468(48)3273
⑥その他広告以外の商品も取扱っております。お気軽にお問合せ下さい。

SHARP X68000 EXEショップ

■SHARP21型 カラーディスプレイテレビCZ-210D 定価179,000円 特価59,800円■

《新発売》X68000EXPERT/PRO シリーズ
PERSONAL WORKSTATION

豊富な周辺機器と多彩なソフトで強力バックアップ!
新型X68000を大割引中!!

●X68000EXPERT
(CZ-602C)1MB/FDD×2
定価¥356,000
●X68000EXPERT-HD
(CZ-612C)1MB/FDD×2、40MB/HDD×1
定価¥466,000

〈メインメモリ〉2Mバイト、
〈拡張IOポート〉2ポート、
〈OS〉オリジナルOS
Human68K Ver.2

●X68000PRO
(CZ-652C)1MB/FDD×2
定価¥298,000
●X68000PRO-HD
(CZ-662C)1MB/FDD×2、40MB/HDD×1
定価¥408,000

〈メインメモリ〉1Mバイト、
〈拡張IOポート〉4ポート、
〈OS〉オリジナルOS
Human68K Ver.2

高性能ワープロ+高性能パソコン

●日本語ワープロ「書院28」搭載!
●MS-DOS™V3.1標準装備!
16ビットパーソナルコンピュータ
MZ-2861 +モニタ 1D10
合計価格¥369,800
限定特価! 210,000円

ハンディCOPY KITプレゼント!
シリアルインターフェース
ACアダプター
カラーイメージ・エディタ
操作マニュアル
〈標準価格¥49,800の品〉

X68000下取りします。CZ662CをCZ600C下取りで差額¥175,000/CZ612CをCZ601C下取りで差額¥225,000

富士通FM-TOWNSセット大特価ご奉仕!!
——AIBITオリジナル・特選セット——

Aセット ①本体/FMTOWNS-1 ②CRT/FMT-DP531 ③キーボード/FMT-KB101 ④OS/TOWNSシステムソフトウェア-V1.1 ⑤本体増設/内蔵マイクロFDドライブ ⑥OS/MS-DOSエミュレータV1.1
①〜⑥計　標準価格¥478,000
ご奉仕大特価¥398,000

Bセット ①本体/FMTOWNS-2 ②CRT/FMT-DP531 ③キーボード/FMT-KB101 ④OS/TOWNSシステムソフトウェア-V1.1 ⑤グラフィックツール/TOWNS PAINT V1.1 ⑥OS/MS-DOSエミュレータV1.1
①〜⑥計　標準価格¥538,000
ご奉仕大特価¥448,000

店頭展示商品を超大特価でおゆずりします。
少数のためTELでお問い合わせください。

アイビット推奨ディスプレイ

●富士通ゼネラルDM405
(14型)
(2000アナログ21/8ピン)
定価¥67,800⇨
特価¥36,000

DM405対応パソコン機種:MSX2。X1シリーズ。MZ700/1500/2000/2200シリーズ。FM77AV/7/8シリーズ。(ケーブルは各専用のものを使用)

●シャープCZ-830D・BK
(14型)
2モードオートスキャン方式
(アナログ/デジタル)
定価¥98,000⇨**大特価**

CZ-830D対応パソコン機種:CZ880C/881C。X1/TURBOシリーズ。ケーブルは本体付属を使用。PC88VA/VA2/VA3/MK2SR/TR/FR/MR。PC9801U/UV/UX/VM/VX/LV各シリーズ。アナログ25ピン↔25ピンケーブルを使用(デジタルは各専用ケーブルで)。MZ700/1500/2000/2200/2500各シリーズ(推奨品シャープ8D8K)。

●シャープCZ-602D
〈ドットピッチ0.39mm〉
(15型アナログTV/3モードオートスキャン方式)
定価¥99,800→**大特価**

●シャープCZ-612D
〈ドットピッチ0.31mm〉
(15型アナログTV/3モードオートスキャン方式)
定価¥119,800→**大特価**
いずれもチルトスタンド付き

●シャープCZ-611D-GY
(15型アナログTV/3モードオートスキャン)
¥145,000→**大特価**

CZ-611D対応パソコン機種:※X1シリーズ/※X1 turboシリーズ/X1 yurboZシリーズ/X68000シリーズ/PC8801シリーズ/PC-9801シリーズ/PC-286シリーズ
(※は接続ケーブルAN1506が必要です)

●シャープCu21CD(21型)
マルチスキャン方式
(アナログ)
定価¥139,800⇨**特価**
特価

CD21CD対応パソコン機種:CZ880C/881C/600C/611C。PC88VA/VA2/VA3/MK25R/TR/FR/MR。PC8801FH/MH/FA/MA。PC286U/V/L。PC9801U/UV/UX/VM/VX/LV各シリーズ。ケーブルは付属を使用(X/シリーズはAN1506で使用)MZ700/1500/2000/2200/2500はAN1508で。

●三菱XC-1498C
(14型アナログ/ドットピッチ0.28mm)
定価¥99,800⇨
特価¥59,800

XC-1498C対応パソコン機種:
NEC・PC9801シリーズ。
エプソンPC286/386シリーズ。

本体
●シャープCZ-822C CP/M付……………¥29,800
●シャープCZ-888C-BK(X1 turbo ZIII)………新発売
●シャープMZ-2520…………¥159,800⇨¥78,000
●NEC PC-9801VX4………¥643,000⇨¥360,000
●NEC PC-9801XA2………¥695,000⇨¥149,000
●NEC PC-98LT11…………
●NEC PC-9801LV21………¥345,000⇨¥240,000
●富士通FM-AV77………¥128,000⇨¥45,000
●富士通FM-AV772………¥158,000⇨¥55,000
●富士通AM-AV40…………¥228,000⇨¥95,000

拡張機器他
●シャープCZ-8TM1(X1用通信ソフト付)‥¥29,800⇨¥6,500
●シャープMZ-1E29(MZ)……¥17,800⇨¥9,800
●シャープX1用ジョイカード……………………¥1,500
●シャープCZ-8EP(I/Oポート)…¥11,800⇨¥9,000
●シャープCZ-8EB3(I/Oボックス)‥¥33,800⇨¥28,000
●シャープMZ-1U09…(2500)…¥9,000⇨¥7,200
●シャープCZ-8BK3…(X1)…¥13,800⇨¥11,700
●シャープCZ-8BK4…(X1)……¥6,800⇨¥5,700
●シャープMZ-1M03…(5500)・¥69,000⇨¥35,000
●シャープMZ8BC04‥(2000)‥¥18,000⇨¥8,000
●シャープMZ-8B104‥(2000)・¥45,000⇨¥18,000
●シャープMZ-1R09…(5500)・¥35,000⇨¥25,000
●シャープMZ-1R10…(5500)・¥30,000⇨¥12,000
●シャープMZ-1R11…(5500)・¥80,000⇨¥40,000
●シャープMZ-1R24…(MZ)…¥22,000⇨¥6,000
●シャープMZ-1R26A‥(2500)・¥13,000⇨¥12,800
●シャープMZ-1R27A‥(2500)・¥13,000⇨¥10,000
●シャープMZ-1R28A・(2500) ¥13,000⇨¥10,000
●シャープMZ-1R29A‥(2500)・¥32,000⇨¥10,000
●シャープMZ-1R37…(2500)・¥35,800⇨¥28,000
●シャープMZ-1T02…(2000)‥¥19,800⇨¥8,500
●シャープMZ-1T03…(1500)‥¥12,000⇨¥8,500
●シャープCZ-8BGR2・(X1)……¥14,800⇨¥4,000
●シャープCZ-8BS1…(X1)…¥23,800⇨¥19,500
●シャープX1、MZ用マウス……………特価¥4,800
●シャープMZ-1X29…………¥13,800⇨¥11,000
●シャープMZ-3500キーボード……………¥10,000
●シャープMZ-5500キーボード……………¥10,000
●シャープX1シリーズ用キーボード…………
●シャープ2000/2200キーボード…………¥10,000
●シャープCZ-64H(ハードディスク)〈CZ-602・652内蔵用〉…¥120,000
●シャープCZ-8NJ2(インテリジェントコントローラー)¥23,800⇨大特価
●富士通MB-22436(FMZマウス)…¥33,000⇨¥19,800
●富士通16βキーボード……¥25,000⇨¥20,000

プリンター
●シャープCZ-8PC3…………¥65,800⇨¥52,000
●シャープCZ-8PC4……………¥99,800⇨大特価
●シャープCZ-8PK5…………¥129,000⇨¥59,800
●シャープMZ-1P22(カラー漢字サーマル)‥¥59,800⇨¥50,000
●シャープMZ-1P27………¥268,000⇨¥214,400
●シャープMZ-1P28………¥148,000⇨¥118,400
●シャープ MZ-1P29………¥168,000⇨¥134,400
●シャープ6P-11(カットシードヒート)…¥95,000⇨¥35,000
●シャープCZ-8PD3(X1用)……¥59,800⇨¥16,000
●NEC-NM9700(漢字プリンタ)‥¥163,000⇨¥88,000
●富士通FMPR-201…………¥79,800⇨¥45,000
●富士通FMPR-351………¥250,000⇨¥125,100
●富士通FMPR-353………¥198,000⇨¥115,000

●富士通MB-27409…………¥98,000⇨¥45,000
●富士通MB-27413…………¥90,000⇨¥25,000
●富士通FMPR-201(漢字カラー)…¥79,800⇨¥45,000
●富士通FMPR-201R1(ROM)…¥23,000⇨¥11,000

ディスプレー(カラー)
●富士通FMTV-211(200)……¥185,000⇨¥89,000
●富士通FMTV-152(200)……¥109,000⇨¥58,000
●富士通MB-27343(200)……¥67,800⇨¥35,000
●富士通TV-151、152(在庫処分品あり)…各¥45,000
●NEC PC-KD854(400)………¥89,800⇨¥58,000

ディスプレー(モノカラー)
●シャープ CZ-1D10(400)……¥41,800⇨¥25,000
●NEC PC-8050(200)…………¥29,800⇨¥24,000

フロッピーディスク
●シャープCZ-503F…………¥49,800⇨¥34,000
●シャープ CZ-503F(インターフェースカードなし)……¥30,000
●シャープCZ-502F…………¥99,800⇨¥75,000
●シャープCZ-300F(CZ-3PCM付)…………¥13,000

ソフト
●ユーカラK2+………(2500)・¥28,000⇨¥23,000
●春望クリエイティブII・(2500)・¥34,800⇨¥29,000
●ビジレス……………(2500)・¥48,000⇨¥42,000
●Hu-CAL日本語……(2500)・¥45,000⇨¥30,000
●ぷりんとしょっぷ……(2500)…¥9,800⇨¥5,000
●G.EDIT2500…………(2500)…¥8,000⇨¥7,000
●FILE UTILITY UT-25F・(2500)…¥6,800⇨¥6,000
●パーソナルCP/M6Z001……¥16,800⇨¥14,800
●V2BASIC6Z010……(2500)‥¥10,000⇨¥8,500
●FORTRAN(1P1213)‥(2500)・¥13,800⇨¥11,700
●C MZ2500 1P1214‥(2500)・¥13,800⇨¥11,700
●COBOL 1P1215……(2500)・¥13,800⇨¥11,700
●C CZ116LF(X1)……………¥13,800⇨¥11,700
●COBOL CZ118LF‥(X1)…¥13,800⇨¥11,700
●ランゲージマスターCZ128SF…¥9,800⇨¥8,500
●シャープCZ-130F(ターボCP/M)……¥14,800⇨¥12,500
●シャープX1・3インチCP/M……¥16,800⇨¥5,000
●富士通B273D030(サウンドエディタ)………¥9,800⇨¥3,000
●富士通B273D040(ミュージックエディタ)……¥9,800⇨¥3,800
●富士通B273D050(グラフィックエディタ)……¥9,800⇨¥3,000
●HUMAN68K CZ-244SS(新発売)‥¥9,800⇨特価

X68000関係ソフト
●マイクロソフトウェアージャパン
「C&プロフェッショナルパッケージ」¥58,000⇨特価
●シャープOS-9/X68000…………¥29,800⇨大特価
●シャープCZ-211LS…………¥39,800⇨大特価!
●シャープCZ-6BE1…………¥35,000⇨大特価!
●シャープCZ-6BE1A…………¥38,000⇨大特価!

SHARPポケットコンピュータ
●PC-1360…………………¥29,800⇨¥19,800
●PCE-200…………………¥22,000⇨大特価
●PCE-500…………………¥28,800⇨¥24,800
●CE-152……………………¥19,800⇨¥9,800
●CE-161プログラムモジュール……¥50,000⇨1個¥3,800
●CE-159プログラムモジュール…………¥35,000⇨¥4,200

ポケコン総合カタログ並びに特価表を差し上げます。
切手¥70を同封の上、当社へお申込みください。
全品新品保証付

本誌発売時には、上記価格よりさらにお求めやすい価格に変更されている場合があります。
上記商品価格には消費税は含まれておりません。
全ての商品に対し、別途3%の消費税金がかかりますのでご了承ください。

☎0426-45-3001〜3
FAX.0426-44-6002
●営業時間/10:00〜19:00●電話受付/20:00迄可●定休日/日曜日(祭日営業)
SHARP SUPER XEX SHOP
アイビット電子株式会社 〒192 東京都八王子市北野町560-5
《夏期休業/8月13日〜15日は勝手ながら臨時休業とさせていただきます。》

信用をモットーに、よりよい品をより安く、迅速にお届けします。
全国通信販売
北海道から沖縄まで
★送料はご注文の際にお問い合わせ下さい。
★掲載の商品は、すべて新品、保証書付きです。
★掲載の商品は充分用意してありますが、ご注文の際は、在庫の確認の上、現金書留または、銀行振込でお申し込み下さい。全商品クレジットでも扱っております。
★お申し込みの際は必ず電話番号を明記して下さい。
★商品、品切れの節はご容赦下さい。
富士銀行八王子支店　(普)1752505

クリエイト特典

- 全商品完全保証書付(メーカー保証)
- 全国無料配達(一部離島の方は有料になります)
- 配達日の指定OK(日曜・祭日にかかわらずお客様のご都合にあわせて配達します)
- どんな商品の組合せも自由自在(ご予算、用途に応じ自由自在にシステムアップできます)
- 中古パソコン高額下取り(今お使いのパソコンをわずかな差額でグレードアップ)
- お支払い方法自由(低金利の均等払い、ボーナス一括払いもご利用ください)

営業時間(年中無休)
AM10:00～PM7:00(日曜・祭日はPM6:00まで)

当社はX68000の販売認定店です。どんなことでも安心してご相談ください。

基本セット X68000 PRO

- CZ-652C(本体・キーボード・マウス) ¥298,000
- CZ-602D(カラー専用ディスプレイ) ¥99,800
- CZ-8PC3(熱転写カラー漢字プリンタ) ¥65,800
- アフターバーナ(ゲームソフト) ¥9,200
- プリンター用紙・ブランクディスケット ¥サービス

■定価合計 ¥472,800

クリエイト特価
電話にてお問合せください。

パソコン通信セット X68000 PRO HD

- CZ-662C(本体・キーボード・マウス) ¥408,000
- CZ-602D(カラー専用ディスプレイ) ¥99,800
- CZ-8PK7(24ピン漢字プリンタ) ¥122,000
- コミュニケーションPRO-68(高機能通信ソフト) ¥19,800
- MD-2400B(オムロン・モデム) ¥49,800
- プリンター用紙・ブランクディスケット ¥サービス

■定価合計 ¥699,400

クリエイト特価
電話にてお問合せください。

格安基本セット X68000 PRO

- CZ-652C(本体・キーボード・マウス) ¥298,000
- CZ-603D(カラー専用ディスプレイ) ¥84,800
- アフターバーナ(ゲームソフト) ¥サービス
- ブランクディスケット(5"2HD・10枚) ¥サービス

■定価合計 ¥382,800

クリエイト特価
電話にてお問合せください。

ミュージックワークセット X68000 EXPERT

- CZ-602C(本体・キーボード・マウス) ¥356,000
- CZ-602D(カラー専用ディスプレイ) ¥99,800
- CZ8PK8(24ピン136桁漢字プリンタ) ¥152,000
- CZ6BM1(MIDIボード) ¥26,800
- MT-32(MIDI音源ユニット) ¥69,000
- AN-S100(アンプスピーカー) ¥36,600
- MUSIC PRO(MIDI版) ¥28,800
- Musicstudio(MIDIマルチレコーディングソフト) ¥25,800
- ブランクディスケット ¥サービス

■定価合計 ¥794,800

クリエイト特価
電話にてお問合せください。

グラフィックワークセット X68000 PRO HD

- CZ-662C(本体・キーボード・マウス) ¥408,000
- CZ-612D(0.31ピッチ・カラーディスプレイ) ¥119,800
- CZ-8NS1(カラーイメージスキャナ) ¥188,000
- CZ-6BN1(スキャナ用パラレルボード) ¥29,800
- CZ-6PV1(ビデオプリンタ) ¥198,000
- IO-730(カラーインクジェットプリンタ) ¥230,000
- Z'sSTAFF PRO-68K ¥58,000
- サイクロンEXPRESS ¥78,000
- CZ-6BE1A(1MB増設RAMボード) ¥38,000
- ブランクディスケット(5"2HD 10枚) ¥サービス

■定価合計 ¥1,347,600

クリエイト特価
電話にてお問合せください。

大サービス・ゲーマーズセット X68000 PRO

- CZ-652C(本体・キーボード・マウス) ¥298,000
- CZ-603D(カラー専用ディスプレイ) ¥84,800
- XE-1PRO(ジョイスティック) ¥サービス
- CZ-8NT1(トラックボール) ¥13,800
- CZ-8NJ2(ジョイカード) ¥23,800
- ドラゴンスピリッツ(ゲームソフト) ¥8,800
- 源平討魔伝(ゲームソフト) ¥7,800
- アフターバーナ(ゲームソフト) ¥9,200
- 沙羅曼蛇(ゲームソフト) ¥8,800
- フルスロットル(ゲームソフト) ¥7,800
- サンダーフォース(ゲームソフト) ¥9,800
- ドッチボール(ゲームソフト) ¥7,800

■定価合計 ¥480,400

クリエイト特価
電話にてお問合せください。

※本広告に掲載の全商品の価格について消費税は含まれておりません。

X68000シリーズ用 周辺機器・ソフトお買い得セール

型番	品名	定価	ソフト名	品名	定価
CZ-6VT1	カラーイメージユニット	¥69,800	MUSIC PRO-68K	マウスを使った楽譜ワープロ	¥18,800
CZ-8NS1	カラーイメージスキャナ	¥188,000	SOUND PRO-68K	サウンドエディタ	¥15,800
CZ-6BE1A	IMB増設RAMボード	¥38,000	Sampling PRO-68K	AD PCMサンプリングエディタ	¥17,800
CZ-6BE2	2MB増設RAMボード	¥79,800	Musicstudio PRO-68K	MIDIマルチレコーディングソフト	¥25,800
CZ-6BE4	4MB増設RAMボード	¥138,000	NEW Print Shop PRO-68K	ポップアートツール	¥19,800
CZ-6BU1	ユニバーサルI/Oボード	¥39,800	Communication PRO-68K	高機能通信ソフト	¥19,800
CZ-6BG1	GP-IBボード	¥59,800	OS-9/X68000	マルチタスクオペレーティングシステム	¥29,800
CZ-6BP1	数値演算プロセッサ・ボード	¥79,800	AI-68K	AI開発ツール	¥188,000
CZ-8NT1	トラックボール	¥13,800	BUSINESS PRO-68K	統合型計算ソフト	¥68,000
CZ-6BM1	MIDIボード	¥26,800	DATA PRO-68K	コマンド型リレーショナルデータベース	¥58,000
CZ-6EB1	拡張I/Oボックス(4スロット)	¥88,000	CARD PRO-68K	カード型リレーショナルデータベース	¥29,800
CZ-6BN1	スキャナ用パラレルボード	¥29,800	TOP財務会計	プロフェッショナル財務会計ソフトウェア	¥200,000
CZ-603D	ドットピッチ0.31mm14型高解像度	¥84,800	Ccompiler PRO-68K	ソフト開発セット	¥39,800
CZ-6TU	パソコンチューナ	¥33,100	Human 68K Ver2.0	開発ツールセット	¥9,800

▲上記以外ビジネスソフト、最新ゲームソフト豊富に在庫あります。※送料はご注文の際お問合せください。●超特価販売中!

★この表以外の組合せ、お支払い方法もご自由にできます。
★X1シリーズ用、X68000シリーズ用各社ハードディスク/プリンタ等の周辺機器を大特価にて販売しております。
電話にてお問合せください。

総合お問合せ先☎03-486-6541(代)

パソコン専門ショップ
ソフトクリエイト 渋谷/横浜

●渋谷店☎03-486-6541(代)
〒150:東京都渋谷区渋谷1-12-7 三和渋谷ビル
振込銀行:三井銀行 渋谷宮益坂支店(普)No.5000340

●横浜店☎045-314-4777(代)
〒221:横浜市神奈川区鶴屋町2-12-8 第1建設ビル
振込銀行:三和銀行 横浜駅前支店(普)No.310852

ACCESS

X1エミュレータ

定価¥9,800

X1エミュレータはX68000上でX1シリーズのアプリケーションを実行するためのソフトエミュレータです。X1のアプリケーションを完全にソフトウェアのみでエミュレートしているため、X1上での実行速度と比較して、平均3〜5倍程度おそくなりますが、X68000のマシン上に実現した仮想X1マシンを楽しめます。また、X1とX68000の相互間でファイルを転送するためのユーティリティと専用ケーブルが付属しますので、X1上で作り上げたソフトの資産をX68000上に移行することも簡単にできます。

ファイル転送ユーティリティ

ディスク転送

X1ディスク⟷X68000 Human68k（5"2Dディスクイメージファイル）

- X1エミュレータではHuman68k上のディスクイメージファイルを仮想ドライブとして使用。

ファイル転送

X1 BASIC：CP/M⟷X68000 Human68k

- X1で作ったプログラム&データをX68000上で使用。

※付属の専用ケーブルをX1とX68000に接続してファイルを転送します。

実行可能アプリケーションソフト

●HuBASIC ●X1 CP／M ●X1 LOGO

【X1 CP／M用ランゲージシリーズ】●APL ●LISP ●COBOL ●C ●FORTH ●FORTRAN ●PASCAL

●etc（X1シリーズ用とされているものに限ります。）

＊プロテクトの施してあるソフトは実行できません。
＊一部サポートしていない機能があります。
＊タイミング等ハードウェアに依存するようなものは、原理上実行できない、もしくは正常に動作しない場合がありますのでご注意ください。
＊turbo専用のソフトは動作致しません。

MS-DOS エミュレータ CONCERTO-X68K

定価¥99,800

CONCERTO-X68K（コンチェルト）はX68000上でお使い頂くMS-DOSエミュレータ（専用ハード+ソフトウェア）です。特定機種用と限定されていないMS-DOS（V2.11）用のソフトがX68000上でお使い頂けます。MS-DOSソフトの実行は、NEC V30CPUを使用した専用ハードウェア（DOS Engine）を利用するため高速実行を実現しております。
ベンチマークテストの結果を見て頂いてもわかるように、PC-9801上で実行するよりもX68000上で実行する方が高速に処理できることを確認しております。

MS-C（4.00）を用いてベンチマークテスト

マシン：X68000 ACEHD　：PC-9801VM（V30）
比較条件：CONCERTO-X68K　：MS-DOS V2.11
　フロッピーディスクを使用：フロッピーディスクを使用
実験方法：FILES＝20
　CONCERTO-X68K側ではMS-DOS V2.11に含まれるCOMMAND.COM上よりコンパイラを起動
　^Cを入力しバッファをクリアした後バッチジョブを実行

＊実験マシンは共にRAMDISK、8087等は使用しておりません。

◆SAVAGE.C

（三角関数、対数関数、平方根関数の演算速度と精度をテストするためのプログラム）

	CONCERTO-X68K	PC-9801VM(10MHz)	PC-9801VM(8MHz)
コンパイル時間+LINK	93	175	174
実行時間	77	78	96

◆SIEVE.C（エラトステネスのふるいプログラム）

	CONCERTO-X68K	PC-9801VM(10MHz)	PC-9801VM(8MHz)
コンパイル時間+LINK	67	119	121
実行時間	116	119	148

（単位は秒　時間計測用プログラムを含む）

```
A>XDOSINIT↲ -------  エミュレータ起動時に必要な初期設定
                      通常はじめに1回だけの実行で可
CONCERTO-X68K  Ver 1.00  Copyright (C) 1988 ACCESS CO.,LTD.

 アドレス 00BE0000 に使用できるDOS Engineがあります。  --
                                                          ¦
 CONCERTO-X68Kを初期設定中です。                          ¦
                                                          ¦
  使用可能なメモリサイズは 512 キロバイトです。    共有メモリ,ハンドシェイク
  DOS Engineからの割り込みレベルは 2 です。        割り込み等のチェック
  8087は実装されていません。                              ¦
                                                          ¦
 CONCERTO-X68Kが使用可能です。                       ----

A>XDOS 〈コマンド〉〈パラメータ〉↲ -------  コマンドはMS-DOSソフト名、パラメータはそのソフトが
                                            必要とするパラメータの並び
 または                                     実行終了後、制御はHuman68kに戻る
A>XDOS COMMAND↲ ---------  COMMAND.COM起動後はMS-DOSの環境として使用可

Command バージョン 2.11

XDOS:A>〈コマンド〉〈パラメータ〉↲ ---------  実行終了後も制御はそのまま

XDOS:A>EXIT↲ -----------  CONCERTO-X68Kを抜けてHuman68kに戻る
```

（CONCERTO-X68Kの実行、下線部はキー入力）

専用ハード：DOS Engine

- 8MHzのV30を使用（メモリノーウェイト）
- ボード上にMS-DOSの実行用メモリ512KByte搭載
- 数値演算プロセッサ8087-1実装可能（オプション）

＊ボードは本体より12cm程度大きくなります。その部分にはカバーがつきます。

MS-DOS用実行可能アプリケーションソフト

- MS-C（Ver 3.00、4.00）
- MS-FORTRAN（Ver3.13、4.01）
- MS-PASCAL（Ver3.13）
- MS-LINK（Ver2.01、2.20、2.44）
- MS-BASIC（Ver5.27）
- Lattice C（Ver2.12、3.10）
- Optimizing-C（Ver2.20F）
- TURBO PASCAL（Ver2.00B、3.01A）
- Plink 86（Ver1.46）
- etc……

（実行可能ソフトの一例です。）

代理店募集

アクセスではこれらの製品の発売にあたり代理店を募集しております。詳しくはお問い合せください。

＊この商品の価格には消費税は含まれておりません。
＊MS-DOSはマイクロソフト社，CP/Mはデジタルリサーチ社の商標です。
COMMAND.COMはMS-DOSに標準のコマンドプロセッサです。上記のソフトウェアは各社の商標です。
＊製品の仕様、名称は予告なく変更する場合がございますのであらかじめご了承ください。

有限会社アクセス
〒101 東京都千代田区神田神保町1-64
神保町協和ビル7F
☎03（233）0200㈹ FAX.03（291）7019

資料請求券
oh!X
8月号

君の味方だ！パソコン通信

ご入会はスタータキットで
買ったその日からアクセスできます。

■申込先
〒556 大阪市浪速区日本橋5-6-7 上新電機株式会社
J&P HOT LINE事務局宛 TEL.(06)632-2521

■利用料金について
入会金/3,000円(スタータキット購入の代金から充当されます)
接続料/3分あたり20円(アクセスポイントまでの電話代は含みません)
※消費税3%が加算されます。

スタータキット申込書

お名前	
お電話番号	
ご住所	〒

お申込品 スタータキット(ソフトなし)
3,000＋90(消費税3%)＝¥3,090

●パソコン/ワープロ通信ネットワークサービス

J&P HOT LINE

アクセスポイントは全国に90ヵ所。日本全国を網羅する、本格的な通信ネットワークです。

スタータキットのお求めは、J&P各店でどうぞ。

店名	住所	電話
渋谷店	東京都渋谷区道玄坂2丁目28番4号	☎(03) 496-4141
町田店	東京都町田市森野1丁目39番16号	☎(0427)23-1313
八王子店	東京都八王子市旭町1番1号八王子そごう7F	☎(0426)26-4141
立川店	東京都立川市幸町4－39－1	☎(0425)36-4141
富山店	富山市双台町1番地	☎(0764)42-2131
大須店	名古屋市中央区大須4丁目2-48	☎(052)262-1141
テクノランド	大阪市浪速区日本橋5丁目6番7号	☎(06) 634-1211
メディアランド	大阪市浪速区日本橋5丁目8番26号	☎(06) 634-1511
コスモランド	大阪市浪速区難波中2丁目1番17号	☎(06) 634-3111
ワープロランド	大阪市浪速区日本橋4丁目9番15号	☎(06) 634-1411
ビジネスランド	大阪市北区梅田1-1-3大阪駅前第3ビルB2	☎(06) 348-1881
阪急三番街店	大阪市北区芝田1-1-3 阪急三番街B1	☎(06) 374-3311
高槻店	高槻市高槻町11番16号	☎(0726)85-1212
くずは店	枚方市楠葉花園町15番2号	☎(0720)56-8181
千里中央店	豊中市新千里東町1-3-204千里サンタウン3F	☎(06) 834-4141
摂津富田店	高槻市大畑町24－10	☎(0726)93-7521
寝屋川店	寝屋川市緑町4－20	☎(0720)34-1166
藤井寺店	藤井寺市岡2丁目1番33号	☎(0729)38-2111
岸和田店	岸和田市土生町2451－3	☎(0724)37-1021
郡山インター店	大和郡山市横田693－1	☎(07435)9-2221
さんのみや1ばん館	神戸市中央区八幡通3-2-16	☎(078)231-2111
京都寺町店	京都市下京区寺町通仏光寺下ル恵美須之町549	☎(075)341-3571
京都近鉄店	京都市下京区烏丸通七条下ル東塩小路町702	☎(075)341-5769
姫路店	姫路市東延末1丁目1番住友生命姫路南ビル1F	☎(0792)22-1221
和歌山店	和歌山市元寺町4丁目4番地	☎(0734)28-1441
奈良1ばん館	奈良市三条町478－1	☎(0742)27-1111
西宮店	兵庫県西宮市河原町5－11	☎(0798)71-1171

ADVANCED

先駆の"Z"アビリティがパソコンクリエイターを魅了する。

TURBO

パソコンテレビ X1 turbo Z III

品名	型名		価格
パーソナルコンピュータ＋キーボード＋マウス	CZ-888C-BK	標準価格	169,800円(税別)
14型カラーディスプレイテレビ	CZ-860D-BK	標準価格	92,200円(税別)
チルトスタンド	CZ-6ST1-B	標準価格	5,800円(税別)

クリエイティブマインドを刺激するAV機能 テレビ、ビデオ、ビデオディスクなどの映像を最大4,096色のリアルな画像で瞬時にグラフィック画面に取り込めるカラー画像デジタイズ機能を標準装備。4段階の量子化取り込み、42通りのモザイク取り込みなど多彩なトリック取り込み処理もサポート。さらにクロマキー合成、インターレーススーパーインポーズ、4,096色対応デジタルテロッパ機能、ステレオFM音源…先駆のAV機能がアートワークの領域をさらに拡げます。

AV指向の高水準ベーシックZ-BASIC搭載 多色グラフィック、カラー画像処理、ステレオFM音源、バンクメモリ対応など、ターボZシリーズが本来もつクリエイティブな機能をフルサポート。また豊富な画面モードで多色を駆使するときに便利なグラフィック用関数（HSV, RGB, HALF, CDOWN, CUP）も装備。さらにFM音源制御用ステートメントとしてX68000と命令コンパチの拡張MMLの採用によりスムーズな8音同時演奏を実現しています。

●メインメモリ128Kバイト標準装備、Z-BASICで最大576Kバイトまでサポート●1Mバイトの5インチフロッピーディスクドライブ2基搭載●JIS第1/第2水準準拠漢字、「システム・ユーザー辞書」を標準装備した高度な日本語処理機能●ニューデザインのマウス標準装備●X1ターボシリーズの豊富なソフト資産が活用できるコンパチブル設計●プリンタ、RS-232Cなど豊富なインターフェイスを装備●ドットピッチ0.39mmのハイコントラストブラウン管、15kHz/24kHzのデュアルスキャン方式採用14型カラーディスプレイテレビ(別売)。

資料請求券
X1 turbo Z III
oh!X
8係

シャープ株式会社 ●お問い合わせは…シャープ㈱電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)
電子機器事業本部テレビ事業部第4商品企画部 〒162 東京都新宿区市谷八幡町8番地 ☎(03)260-1161(大代表)

本広告に掲載しております商品および役務の価格には消費税は含まれておりませんので、ご購入の際、消費税をお支払い下さい。

T4910217908562 雑誌02179-8